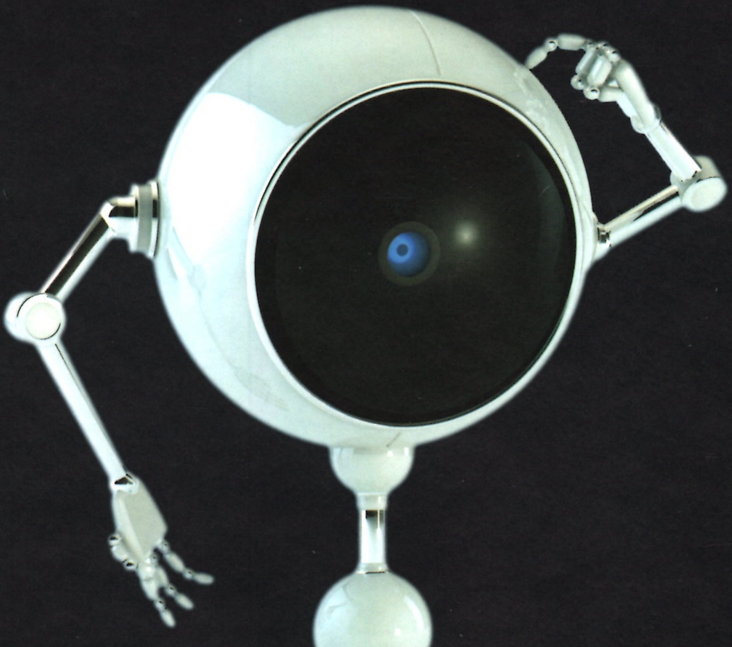
LA COMPUTACIÓN

TURING

Pensando en máquinas que piensan



NATIONAL GEOGRAPHIC

ALAN TURING recibió en 2009, sesenta y cinco años después de su muerte, las disculpas del gobierno del Reino Unido por el modo en que fue tratado en vida. Declarado culpable de actos homosexuales y obligado a seguir un tratamiento químico que le provocó impotencia, se suicidó a la edad de 41 años. Se truncó así la carrera de una de las figuras clave en el desarrollo de la computación: amén del primer modelo de funcionamiento de un hipotético ordenador con unidad central de proceso, la conocida como «máquina de Turing», contribuyó a la construcción de algunos de los primeros ingenios computacionales de la historia y se valió de ellos para descifrar los códigos militares nazis, una empresa cuyo éxito salvó incontables vidas y aceleró el final de la guerra. Es la suya, en definitiva, la historia trágica de un genio que fue empujado a la muerte por la nación que tanto hizo por defender.

LA COMPUTACIÓN

TURING

Pensando en máquinas que piensan

NATIONAL GEOGRAPHIC

RAFAEL LAHOZ-BELTRA es profesor de matemática aplicada en la Universidad Complutense de Madrid. Es autor de diversos libros de divulgación sobre la intersección entre biología e informática y entre aquella y la matemática, y en especial acerca de la labor de Alan Turing en este ámbito.

Realización: EDITEC

Diseño cubierta: Llorenç Martí

Diseño interior: Luz de la Mora

Infografías: Joan Pejoan

Fotografías: Age Fotostock: 25, 39ai, 39b, 59ad, 91b; Archivo RBA: 65, 74d, 141; Bletchley Park Museum: 70; Corbis: 59b; CWI/Rascal/Lego: 39ad; Getty Images: 59ai, 69a, 91a, 92, 119a; Simon Harriyott: 21ad; Rafael Lahoz-Beltrá: 33, 47, 56, 85, 111, 139, 142; National Security Agency: 69b; Princeton University: 36; Christian Richardt: 21b; RJB1: 74i; Sergei Frolov/Soviet Digital Electronic Museum: 112; Dr Tony Shaw: 119b; Sherbone School: 21ai; U.S. Army Photo: 89.

ISBN: 978-84-473-7638-4
Depósito legal: B-2618-2016

Impreso y encuadernado en Rodesa, Villatuerta (Navarra)

Impreso en España - *Printed in Spain*

Sumario

Introducción

Pese a su corta vida, Alan Turing fue uno de los personajes más influyentes del siglo xx. Algunos de los hitos de su carrera científica son el diseño de una máquina hipotética, la máquina de Turing, con cuya ayuda creó los conceptos teóricos que permitieron la construcción de los primeros ordenadores, y la confección de uno de los ordenadores más rápidos de su época, el Pilot ACE. Como criptógrafo destacó por conseguir desvelar los códigos Enigma, con los que los alemanes cifraban sus mensajes durante la Segunda Guerra Mundial. Y además realizó investigaciones pioneras con las que sentó las bases de la inteligencia artificial y la biología matemática.

Esta obra tiene como objetivo explicar de una forma amena y rigurosa la naturaleza de estas aportaciones fundamentales para la evolución del mundo contemporáneo.

Bajo este enfoque hemos aunado en un mismo libro elementos propios de un texto de «ciencia recreativa» con otros que son propios de una biografía, demostrando como algunos de los hallazgos más importantes de Alan Turing forman parte de nuestra vida diaria. Así, por ejemplo, el libro da respuesta a algunos interrogantes como ¿qué es un ordenador?, ¿por qué se *cuelgan* los ordenadores?, ¿qué país inventó el ordenador?, ¿resuelven los ordenadores toda clase de problemas?, ¿qué es un *captcha*?, ¿qué es un sistema de reconocimiento óptico de caracteres (OCR)?,

¿puede haber máquinas inteligentes?, ¿cómo funciona un ordenador cuántico?, entre otras preguntas.

El carácter polifacético de las investigaciones de Alan Turing fue una manifestación más de su genialidad. Su capacidad para encontrar nuevas áreas de investigación y relaciones entre fenómenos o cuestiones aparentemente dispares solo fue igualada entre sus contemporáneos por el matemático húngaro John von Neumann. Con estos dos científicos nació en la década de 1940 el «científico multidisciplinar», el sujeto sin fronteras en el conocimiento, capaz de abstraer de la biología, la economía, la sociología o la física los elementos comunes utilizando las matemáticas y los ordenadores, con la finalidad de unificar problemas en apariencia distantes, pero en el fondo similares.

Turing es un personaje cuya vida y obra no dejan indiferente. Su vida como científico fue una auténtica aventura intelectual, rica en matices y hallazgos, y su vida privada, marcada por su homosexualidad en una época en la que en Gran Bretaña era considerada delito penal, estuvo repleta de anécdotas que lo convierten en un personaje singular, con una personalidad alejada de lo común. Los problemas derivados de su orientación sexual le provocaron la segunda de las profundas depresiones que padeció y que le llevó a acabar con su vida mediante la ingesta de cianuro, si bien el misterio que rodeó su muerte ha dado pie a otras conjeturas, entre ellas la del asesinato.

Esta obra, que descubre tanto al hombre como al científico, se organiza en cinco capítulos. En el primero, tras un recorrido biográfico por su infancia y juventud hasta concluir sus estudios en la Universidad de Cambridge, se describe detalladamente una de sus principales contribuciones científicas: la máquina de Turing, con las distintas variantes diseñadas por el genio británico y también por otros investigadores. Se describen también algunas experiencias sobre la construcción de máquinas de Turing o su simulación a través de *software*. El capítulo concluye con cuestiones más concretas, como el problema de la parada, que explica, entre otras cosas, por qué se «cuelga» un ordenador.

En el segundo capítulo se narra cómo el acoso alemán sufrido por el Reino Unido durante la Segunda Guerra Mundial llevó a los

británicos a crear Bletchley Park, lugar donde los criptógrafos, entre ellos Turing, lograron finalmente descifrar los mensajes interceptados a los alemanes del III Reich. En ese contexto bélico, personajes como Turing, entre otros muchos, desplegaron todo su talento, recibiendo un merecido reconocimiento al final de la contienda. Fue precisamente en Bletchley Park donde vio la luz el Colossus, considerado hoy el primer ordenador digital de la historia. La Segunda Guerra Mundial no solo fue un derroche en vidas, sino también de inteligencia. Tras esta estimulante experiencia, Alan Turing fue capaz de dar el salto definitivo desde el mundo abstracto de la máquina que lleva su nombre hasta el mundo real, construyendo un ordenador: el Pilot ACE.

El tercer capítulo aborda una cuestión cuya polémica sigue vigente: ¿quiénes inventaron el ordenador, los británicos o los estadounidenses? Según la última revisión histórica fue el Reino Unido, gracias al Colossus, el país merecedor de este reconocimiento, por la fecha en que fue construido y por ser en aquella época el país más avanzado en el diseño y construcción de ordenadores. Entonces ¿por qué Estados Unidos le arrebató esta industria?

Una vez descritas las características del Pilot ACE, y tras dar una respuesta a estas preguntas, nos adentraremos en la arquitectura de Von Neumann, esto es, la forma en que desde entonces se organizan a nivel lógico y funcional los componentes de un ordenador, para concluir con los años en que Alan Turing se dedicó a la programación de ordenadores en la Universidad de Manchester.

Ya próximo el final de su vida, Turing puso el colofón con lo que tal vez sea uno de sus proyectos más ambiciosos, sentando las bases teóricas de lo que más adelante se llamaría inteligencia artificial. Continuó su labor en la Universidad de Manchester, aunque en esta ocasión dio un paso más y se hizo una pregunta realmente ambiciosa: ¿puede haber máquinas inteligentes?, materia de la que trata el cuarto capítulo. Turing diseñó circuitos de neuronas artificiales y creó una prueba aún en vigor, el test de Turing, con el que evaluar si una máquina, por ejemplo un ordenador, se comporta o no de modo inteligente cuando juega al ajedrez, traduce un texto de un idioma a otro o realiza cualquier otra tarea para la que un ser humano utilizaría su inteligencia.

La última etapa de su vida fue tan fértil científicamente como la primera. Fue en sus últimos años cuando utilizó por vez primera un ordenador para el estudio y simulación de problemas biológicos, elaborando modelos matemáticos sobre el crecimiento y la formación de patrones en los seres vivos, al intentar dar respuesta a la cuestión de cómo se forman los patrones de bandas en la piel de las cebras. Fruto de estos estudios es el nacimiento de una nueva disciplina, la biología matemática. En la primavera de 1954, Alan Turing puso fin a su vida tras ingerir una manzana envenenada a la edad de cuarenta y un años.

Este libro concluye con un quinto capítulo en el que se detalla su legado científico. Por razones obvias se ha omitido cualquier referencia a los ordenadores actuales, ya sean superordenadores, personales de mesa, portátiles, *netbooks*, *tablets* o cualquier otro formato, y también a aquellos dispositivos en los que subyace un ordenador, ya sea un teléfono móvil, agenda personal o cualquier otro. Todas estas máquinas no son más que la evolución natural de la máquina teórica de Turing y de los primeros ordenadores, Colossus, ENIAC, Pilot ACE, EDSAC, y un largo etcétera hasta el momento actual. El legado de Turing no solo fueron sus aportaciones científicas, sus geniales hallazgos y su contribución a la informática, sino lo que su trabajo sugirió pero dejó sin concluir, y que ha inspirado a generaciones posteriores de científicos. Como ejemplo se describen tres proyectos o líneas de investigación aún abiertas y en pleno desarrollo: el ordenador cuántico, el diseño de modelos de redes neuronales artificiales y su utilización en sistemas inteligentes en la vida diaria, y el estudio del ADN mediante ordenadores, la molécula de la vida, cuya estructura fue descubierta en el Reino Unido por Watson y Crick un año antes de su muerte.

Todo un viaje apasionante guiado por una de las mentes más interesantes y geniales del siglo XX, un pensador de máquinas pensantes, la fascinación de cuya figura no hace sino aumentar más de medio siglo después de su fallecimiento.

1912 El 23 de junio nace en Londres Alan Mathison Turing. Es el segundo hijo de Julius Mathison Turing y Ethel Sara Stoney.

1926 Tras aprobar el examen de ingreso a la escuela privada, es aceptado en el Sherborne School.

1931 Ingresa en el King's College de la Universidad de Cambridge, donde estudia matemáticas.

1935 Obtiene una beca por dos años para trabajar en el King's College.

1936 Inicia los estudios de doctorado en la Universidad de Princeton, en Estados Unidos, que acabará en 1938. Rechaza una oferta de trabajo de Von Neumann para trabajar en Princeton y regresa al King's College. Introduce la noción de máquina de Turing, uno de los conceptos claves de la computación.

1939 Se incorpora como criptógrafo al complejo de Bletchley Park. Inventa Bombe, la máquina con la que los británicos lograron romper con éxito los códigos alemanes Enigma.

1945 Recibe la Orden del Imperio Británico en reconocimiento a su contribución como criptógrafo a la victoria de los británicos en la Segunda Guerra Mundial. Se traslada al Laboratorio Nacional de Física en Londres para encargarse de la creación del ordenador Pilot ACE, cuyo diseño presentó al laboratorio en 1946.

1948 Se incorpora a la Universidad de Manchester, donde, junto con Max Newman, organiza un laboratorio dedicado al diseño y construcción de ordenadores con fines científicos. Como resultado se crea el Manchester Mark I, reemplazado por el ordenador Ferranti Mark I en 1951. Introduce una de las primeras redes neuronales artificiales.

1950 Se publica el artículo «Computing Machinery and Intelligence», en el que introduce el test de Turing. Se trata de una prueba fundamental con la que evaluar si un ordenador, un programa o una máquina se comportan o no de un modo inteligente. Programa el ordenador MADAM de la Universidad de Manchester para que escriba cartas de amor.

1952 Turing presenta las ecuaciones de reacción-difusión con las que nace uno de los primeros trabajos de biología matemática: el estudio de la morfogénesis. Es arrestado, acusado de homosexualidad y condenado a seguir un tratamiento hormonal para anular su libido.

1954 A los cuarenta y un años Turing se suicida, presuntamente al ingerir una manzana impregnada en cianuro.

CAPÍTULO 1

¿Qué es un ordenador?

Ya en el siglo XVII Blaise Pascal y Gottfried Leibniz inventaron máquinas con las que se podían realizar operaciones aritméticas elementales. Sin embargo, el propio Leibniz albergaba otro sueño, crear una máquina que fuera capaz de razonar. Hubo que esperar hasta el siglo XX para que Alan Turing desarrollara los conceptos teóricos que permitieron la construcción de los primeros ordenadores.

El 23 de junio de 1912 nacía en Londres Alan Mathison Turing. Hasta un año antes sus padres, Julius Mathison Turing y Ethel Sara Stoney, residían en Chatrapur, India, ciudad en la que por aquel entonces su padre era empleado del Indian Civil Service. Tras ser concebido, Julius y su esposa pensaron que sería mejor que su futuro hijo naciera en el Reino Unido, así que viajaron a Londres para que Sara diera a luz a Alan, el segundo y último hijo del matrimonio Turing. Después del nacimiento de Alan, su padre consideró que la India podría ser un lugar peligroso para vivir, así que dejó instalada a su familia en Inglaterra mientras él se dedicaba a desempeñar su labor de funcionario en Chatrapur, y realizaba varios viajes de ida y vuelta entre la India y el Reino Unido. Tras cumplir Alan el primer año, Sara viajó también a la India para reunirse con su marido, dejando a sus dos hijos al cuidado de un matrimonio de confianza. En esa época sus padres viajaron en numerosas ocasiones entre Guildford, a las afueras de Londres, y la colonia británica.

Los padres de Turing pertenecían a la clase media-alta de una sociedad que aún conservaba las costumbres y los valores educativos tradicionales del Imperio británico, algo muy alejado de la personalidad de Alan y contra lo cual chocaría a la postre. Tanto su familia paterna como la materna carecían de tradición científica o académica. El único Turing que adquirió cierta fama, Harvey

Doria Turing, hermano de Julius, lo hizo gracias a su habilidad en la pesca con mosca. Sin embargo, pese a la ausencia de tradición intelectual en su familia, Alan manifestó ser una mente muy despierta a una temprana edad. Se cuenta que desde pequeño manifestó un gran interés por los números, las letras y los rompecabezas, y por ejemplo, mientras paseaba se paraba ante las farolas para observar detenidamente su número de serie. También se cuenta que a la edad de siete años, en una merienda en Ullapool, ciudad situada en el norte de Escocia, el jovencito Alan pensó que la mejor manera de recolectar miel era dibujar el trayecto que realizaban las abejas para obtener así el punto de intersección de los recorridos, dado que ese lugar marcaría precisamente dónde se encontraba el panal. En otra ocasión se percató de que la cadena de su bicicleta se salía tras un número determinado de vueltas del plato y los piñones; al parecer le atraía más la idea de resolver un problema que comprar una nueva cadena, opción que hubiera preferido cualquier niño de su edad.

Desde su niñez, Alan Turing mostró pasión por la ciencia, de la que daría signos durante sus años de escolarización. A la edad de seis años, su madre lo matriculó en St. Michael's, un colegio público en el que daban especial importancia a la enseñanza del latín. Allí comenzaría su formación dentro del sistema educativo inglés, con el que tendría sus más y sus menos a lo largo del tiempo, pues si bien es verdad que es la fuente de la que bebió y que a la postre lo conformó intelectualmente, también es cierto que, dada su personalidad, entró en conflicto con él por sus valores especialmente clasistas, que se sustentaban fundamentalmente sobre dos pilares: la Iglesia anglicana y el Imperio británico. Ese ambiente inculcó en Alan uno de sus rasgos característicos: su respeto hacia las normas. Una anécdota ilustra bien este aspecto de su personalidad: un día que su madre le estaba leyendo *The pilgrim's progress* (*El progreso del peregrino*; 1678), una de las novelas clásicas de la literatura inglesa escrita por John Bunyan (1628-1688), se saltó una parte porque pensó que era pesada y aburrida por su contenido religioso para un niño, y entonces Alan le hizo ver a su madre que la parte que se había saltado era esencial y sin su lectura la historia que le estaba leyendo carecía de sentido.

Concluida la etapa en el St. Michael's, siguió la misma trayectoria que su hermano mayor, John. Primero ingresó en el centro Hazelhurst, y a continuación fue inscrito en su primer colegio privado, el Marlborough. Por aquella época, como otros muchos niños, Alan realizó experimentos elementales de química y se interesó por la lectura de un libro muy popular en la época, titulado *Natural wonders every child should know (Maravillas de la naturaleza que todo niño debería conocer)*, de Edwin Tenney Brewster (1866-1960). Este libro fue sin duda de gran importancia para él, ya que le abrió los ojos al modo en el que el científico interroga y explica la naturaleza, además de que fue la primera vez que el joven Turing leía algo relacionado con la biología en cuya explicación se utilizaba la idea de «máquina»: la obra enunciaba que el cuerpo humano era una «compleja máquina» cuya principal misión era el mantenimiento de la vida.

> «La idea detrás de los computadores digitales puede explicarse diciendo que estas máquinas están destinadas a llevar a cabo cualquier operación que pueda ser realizada por un equipo humano.»
>
> — Alan Turing, «Maquinaria de computación e inteligencia».

Las matemáticas, la química y curiosamente el estudio del francés llamaron sin duda su atención. Su madre lo inscribió en el Hazelhurst Preparatory School, donde pese a ser un buen estudiante, no destacó en demasía, más bien fue un alumno discreto dentro de la media general. Más adelante, ya en el Marlborough School, su madre tuvo que sacarlo del centro, al parecer por haber padecido algún episodio de acoso escolar. Al contrario de lo que esta circunstancia pudiera hacer pensar, Turing gozaba ya entonces de una complexión atlética que conservaría a lo largo de toda su vida. En la Inglaterra de la época las cualidades atléticas eran un factor no menor en una experiencia académica completa, lo cual, unido a los destellos de inteligencia superior de Turing, parecían apuntar al éxito académico. Su madre, no obstante, llegó a dudar de las capacidades de su hijo para recorrer sin problemas

el exigente sistema educativo privado inglés. Que su hijo fuera aceptado en una escuela privada de prestigio fue para ella una verdadera obsesión, ya que ello representaba un distintivo de la clase social a la que la familia pertenecía. Finalmente, en 1926, y pese a los temores maternos, Alan aprobó con éxito el examen de ingreso a la escuela privada, conocido en inglés como Common Entrance Examination, siendo aceptado por fin en el Sherborne School.

Los años de formación en dicho centro fueron decisivos para consolidar su personalidad. Allí mostró una especial inclinación por resolver los problemas que él mismo se planteaba, y no tanto por los temas tratados por sus profesores. Como suele ser habitual, también en la actualidad, el sistema de enseñanza de la época resultaba poco estimulante para los alumnos más brillantes. Alan ganó premios escolares en matemáticas, leyó la teoría de la relatividad de Einstein y se documentó sobre mecánica cuántica gracias al célebre libro de Arthur Eddington (1882-1944) titulado *The nature of the physical world* (*La naturaleza del mundo físico*; 1928). Era tan singular su personalidad que en cierta ocasión el director de la escuela dijo de él:

> Si permanece en la escuela privada, debe tener la intención de ser educado. Ahora bien, si solo pretende ser un científico especialista, entonces está perdiendo el tiempo aquí.

Entre las anécdotas que se recuerdan de él a la edad de catorce años, y que demuestran su carácter tenaz y perseverante, está la que tuvo lugar durante una huelga general en el Reino Unido en 1926. Pese a la huelga, quiso asistir a clase, siendo tal su determinación que fue capaz de recorrer unos 100 kilómetros en bicicleta en el trayecto que había desde la escuela hasta su casa en Southampton, con parada incluida en una pensión para pasar la noche.

En esta escuela permaneció desde 1926 hasta 1931. Parece ser que las normas tan estrictas de este centro hicieron de él una persona aún más tímida y retraída. Si bien es verdad que sus profesores lo colocaron entre los últimos alumnos de la clase en

griego, latín e inglés, en matemáticas desplegó toda su genialidad, pues fue capaz de obtener la serie infinita de una función trigonométrica, en particular de la inversa de la tangente:

$$\arctan x = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} \ldots$$

En 1928, a la edad de dieciséis años, Alan fue capaz de «entender» la teoría de la relatividad de Einstein, y en 1929 comenzó a leer con gran entusiasmo a Schrödinger y la mecánica cuántica. Fue precisamente durante ese año cuando conoció y entabló una estrecha amistad con Christopher Morcom, un alumno de un curso superior. Se trataba de un muchacho de gran talento científico, que falleció repentinamente dos años después a causa de la tuberculosis. Durante ese corto período Christopher y Alan entablaron una gran amistad, compartiendo sus inquietudes científicas juveniles. Se trató sin duda de la primera vez que Alan Turing conocía a alguien de su edad con inquietudes y gustos parecidos. Además, esta amistad sirvió para que Alan mejorara algunas cualidades personales, por ejemplo, se volvió más comunicativo con otras personas. Su amistad fue tal que ambos viajaron al Trinity College, en Cambridge, para solicitar dos becas que les permitieran estudiar en tan insigne centro. Otra de las muestras de su tenacidad es que en esa época tuvo que examinarse dos veces para conseguir una beca en la Universidad de Cambridge, la primera en 1929, y la definitiva, conseguida con perseverancia, al año siguiente tras presentarse de nuevo. Sin embargo, todos aquellos sueños juveniles de amistad e inquietudes comunes se vinieron abajo tras el regreso a Sherborne. La repentina muerte de su amigo tuvo un gran impacto en su ánimo, sumergiéndolo durante un tiempo en una cierta crisis que lo apartó de la religión, lo que le condujo a una postura próxima al ateísmo. Curiosamente, durante casi tres años, según consta en las cartas escritas por Turing a la madre de Morcom, estuvo preocupado sobre cómo la mente humana, y por tanto la de su amigo fallecido, se alojaba en la materia, es decir en el cuerpo humano. Más aún, y pese a su incipiente ateísmo, creía que la mente sobrevivía al cuerpo y se preguntaba cuál era el mecanismo mediante el que la mente se liberaba definitivamente del cuerpo

tras la muerte. La lectura del libro de Eddington condujo a Alan a plantearse la posibilidad de que la mecánica cuántica tuviera algo que ver con la cuestión, lo que constituye, dada su edad, otra manifestación más de su talento e ingenio, si tenemos en cuenta que esta hipótesis, el papel desempeñado por la mecánica cuántica en el problema clásico de la relación entre mente y materia, es precisamente el fundamento de una de las investigaciones llevadas a cabo por varios científicos a mediados del siglo XX.

«La ciencia es una ecuación diferencial. La religión es una condición de contorno.»

— Alan Turing, en una carta al matemático inglés Robin Gandy.

En 1931 Alan Turing ingresó como estudiante de matemáticas en el King's College de la Universidad de Cambridge, separando su camino del de su hermano mayor, John Ferrier Turing, quien ejerció la abogacía en Londres. Afortunadamente para Alan, la universidad fue un lugar más propicio que los diferentes colegios por los que había transitado, y en Cambridge encontró por fin el ambiente intelectual necesario para desarrollar sus inquietudes. Según el biógrafo de Turing, Andrew Hodges, fue en 1932 cuando Alan Turing admitió la homosexualidad como uno de los rasgos de su personalidad. Al año siguiente, tuvo su primera relación amorosa con un estudiante de matemáticas, James Atkins, quien abandonaría los estudios para iniciar una carrera como músico. Ciertos círculos literarios y asociaciones universitarias eran en esta época lugares idóneos en los que alguien como Turing podía establecer relaciones afectivas. Sin embargo, él siempre evitó dichos ambientes y dedicó parte de su tiempo libre a practicar deportes al aire libre, como correr o remar. En su faceta académica, aunque en 1932 leyó un trabajo de John von Neumann sobre los fundamentos lógicos de la mecánica cuántica, el tema por el que mostró mayor interés fue la lógica matemática. Se sabe que leyó la obra de Bertrand Russell (1872-1970) *Introduction to mathematical philosophy* (*Introducción a la filosofía matemática*; 1919) y el célebre *Principia mathematica* (1910-1913), de Russell



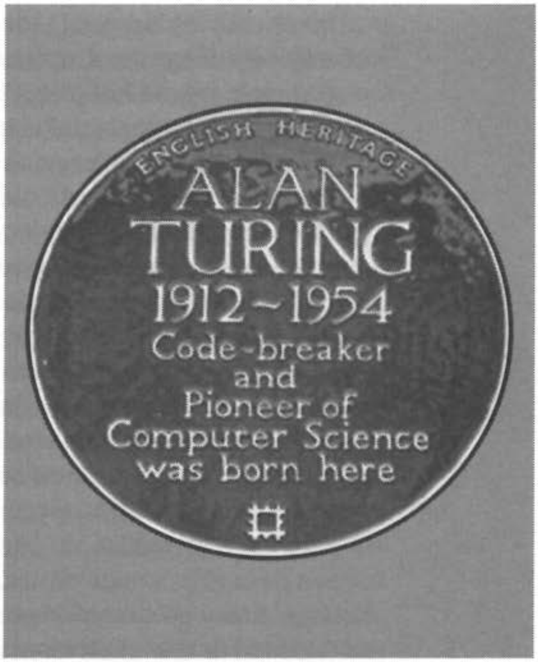




FOTO SUPERIOR IZQUIERDA:
Alan Turing en 1928, a la edad de dieciséis años.

FOTO SUPERIOR DERECHA:
«Alan Turing, 1912-1954, criptógrafo y pionero de la ciencia computacional, nació aquí.» Así reza una de las cinco placas azules repartidas por el Reino Unido que conmemoran los distintos lugares donde vivió Turing.

FOTO INFERIOR:
El King's College, adjunto a la Universidad de Cambridge, donde Turing ingresó en 1931.

y Alfred North Whitehead (1861-1947). No cabe duda de que estas lecturas contribuyeron a su madurez intelectual.

Sin embargo, si hubo una figura científica que tuvo un gran impacto sobre Turing esta fue Kurt Gödel (1906-1978) a través de su famoso artículo publicado en 1931 sobre los llamados *teoremas de incompletitud*. Este artículo fue uno de los motivos que condujo a Turing a idear lo que se conoce como *máquina de Turing*, una máquina de propósito general que de forma automática es capaz de decidir qué funciones matemáticas pueden ser calculadas y cuáles no. Si una función puede ser calculada, entonces la máquina, transcurrido un cierto tiempo que debe ser finito —en palabras de otra figura de las matemáticas, David Hilbert (1862-1943)—, proporcionará un resultado. Por el contrario, si una función no es computable, entonces la máquina realizará cálculos una y otra vez, sin detenerse. Según Hodges, Turing fue más un filósofo que un matemático, lo que explicaría su interés por los problemas de la lógica matemática. Tal vez sin ser consciente de ello Turing contribuyó a crear los fundamentos teóricos de la computación antes de que el ordenador fuese una realidad tangible.

En 1933 Adolf Hitler ascendió al poder en Alemania, un acontecimiento que anunciaba una nueva contienda internacional, la Segunda Guerra Mundial. Alan Turing, preocupado por los acontecimientos políticos y sociales que estaban teniendo lugar en el Reino Unido y en el resto de Europa se unió al movimiento antibélico. No obstante, esta adscripción no significó que fuera marxista o pacifista como muchos otros simpatizantes. Años después, Turing, como millones de personas más, se vio involucrado en la contienda, en su caso ayudando como criptógrafo a su país y al resto de los Aliados a salir victoriosos de la guerra.

LA «MÁQUINA-A» DE TURING

En 1934 Alan Turing concluyó sus estudios en la universidad graduándose en matemáticas. Al año siguiente obtuvo una beca de dos años del King's College, uno de los colegios que conforman la

Universidad de Cambridge. Fue una época de éxitos, en los que Turing dejó entrever algunos destellos de su genialidad. En 1936 ganó el premio Smith —otorgado por la Universidad de Cambridge a jóvenes investigadores en física teórica, matemáticas o matemática aplicada— por su trabajo en teoría de probabilidad, titulado «On the gaussian error function» («Sobre la función de error gaussiana»), que nunca fue publicado. Curiosamente en ese trabajo de investigación redescubrió el famoso «teorema central del límite», uno de los teoremas más importantes de la estadística. Ese mismo año escribió un artículo científico decisivo, titulado «On computable numbers with an application to the Entscheidungsproblem» («Sobre los números computables con una aplicación al Entscheidungsproblem»), en el que hará una de las aportaciones científicas más importantes de su vida: la máquina de Turing. Con el trabajo realizado tras su graduación el futuro académico de Turing estaba asegurado, los primeros pasos de una brillante carrera ya estaban dados.

En la primavera de 1935 Turing había asistido a un curso en el campus de la Universidad de Cambridge, donde, recordemos, trabajó como becario. El curso fue impartido por Max Newman (1897-1984), un insigne topólogo de la época con el que Turing trabó una duradera y franca amistad. La topología es una especialidad de las matemáticas que estudia las propiedades de los objetos que se conservan cuando los transformamos de manera continua. Desde entonces la amistad entre Newman y Turing se mantuvo firme durante toda su vida, siendo especialmente beneficiosa para su trayectoria científica. Max Newman coincidió con Alan Turing en varios episodios de su vida, como por ejemplo, durante la Segunda Guerra Mundial en Bletchley Park en la tarea de descifrar los mensajes interceptados a los alemanes, o tiempo después, en la Universidad de Manchester, donde escribieron programas para Baby, uno de los primeros ordenadores construidos tras la contienda.

En Cambridge, Turing tuvo la ocasión de participar en uno de los capítulos más fascinantes de la matemática. El filósofo y matemático británico Bertrand Russell sostenía que la lógica era un sólido soporte para las verdades matemáticas. Esta idea era pre-

cisamente el núcleo de su libro *Principia mathematica*, escrito tiempo atrás en colaboración con el filósofo Whitehead. Si las matemáticas podían ser interpretadas desde un punto de vista lógico, nada impedía entonces que esta disciplina fuera reducida a los dominios de la lógica. Ahora bien, a principios de los años treinta, otro filósofo y matemático, Kurt Gödel, nacido en Brno (República Checa), por aquel entonces parte del Imperio austrohúngaro, había enunciado un célebre principio filosófico en el ámbito de la matemática. Gödel introdujo lo que se conoce como *teorema de incompletitud*, que puede resumirse en la idea de que hay enunciados matemáticos o proposiciones —los denominados *indecidibles*— que no pueden probarse ni refutarse. En general, una proposición es una afirmación que puede ser verdadera o falsa. Por ejemplo, si alguien dice «2 + 3 = 5» podemos establecer que dicha afirmación es verdadera. En lenguaje propio de la matemática, tendríamos que:

$$A = [2 + 3 = 5] \Rightarrow [A \text{ es verdadero}]$$

Por el contrario, si una persona propone el siguiente producto o multiplicación «2 × 3 = 8», entonces, sin lugar a dudas, diríamos que esta afirmación es falsa:

$$B = [2 * 3 = 8] \Rightarrow [B \text{ es falso}]$$

Sin embargo, hay proposiciones en las que cuando se pretende establecer su veracidad o falsedad se produce lo que se llama una *paradoja*, que consiste en una proposición que se contradice a sí misma. Por ejemplo, cuando el filósofo Sócrates dice «Solo sé que no sé nada» cae en una contradicción, ya que si Sócrates ya sabe que «no sabe nada», entonces «ya sabe algo». Un ejemplo clásico, trasladando una vez más esta situación desde la matemática al lenguaje, es la conocida *paradoja del mentiroso*.

Gödel trasladó esta paradoja del lenguaje a la matemática, en particular al ámbito de la lógica, demostrando en 1931 el llamado *teorema de incompletitud de Gödel*, donde se caracterizan los sistemas incompletos, aquellos en los que no podemos evaluar si

LA PARADOJA DEL MENTIROSO

Supóngase que expresamos de la misma manera que lo haría un matemático la siguiente proposición *G*:

G = [Esta afirmación no es verdad]

Si asumimos que la proposición es verdadera entonces concluiremos que la proposición es falsa. O a la inversa, si decidimos que *G* es falsa, entonces concluiremos que *G* es verdadera. Esta paradoja tiene lugar en los llamados *sistemas autorreferenciales*, tal es el caso de la frase del ejemplo o de forma similar del tipo «Yo estoy mintiendo». Esta situación da como resultado un «bucle extraño». Se trata de situaciones en que independientemente de cómo nos desplacemos, siempre acabamos en el mismo punto, que no es otro que aquel en el que comenzamos. Algunos ejemplos serían una mano dibujando a otra, como en el célebre cuadro de Escher, la síntesis de proteínas y el ADN en una célula, o un «micrófono escuchando su altavoz» según ilustra Douglas Hofstadter en su libro *Soy un bucle extraño*.



***Drawing hands* (1948), obra de Mauritis Cornelis Escher.**

sus proposiciones son verdaderas o falsas. Una cuestión realmente apasionante es cómo estas consideraciones filosóficas, y aparentemente alejadas del mundo real, hicieron tambalearse los cimientos de la matemática. Es en esta época cuando algunos filósofos y matemáticos se formulan la siguiente pregunta: ¿puede la intuición matemática ser codificada en un conjunto de reglas, o, tal como se plantea la cuestión en la actualidad, en un programa de ordenador? Es decir, lo que se pretendía era averiguar si sería posible o no construir algún ingenio mecánico, actualmente un ordenador, con el que pudiéramos averiguar o demostrar de un modo automático, sin la intervención humana, la veracidad o fal-

sedad de alguna demostración o afirmación de índole matemática. Por ejemplo, en lo que hoy se conoce como *razonamiento automatizado* no hay ningún sistema de reglas computacionales o de deducción que nos permita determinar con un programa las propiedades de los números naturales. Los números naturales, $\mathbb{N}=\{1,2,3,4,\ldots\}$, esto es aquellos que usamos para contar los elementos de un conjunto, por ejemplo «número de manzanas», tienen una serie de propiedades.

Considérese el siguiente ejemplo. Sean a, b y c un número de manzanas igual a 2, 3 y 5 respectivamente. La propiedad asociativa establece que $(a + b) + c = a + (b + c)$, mientras que la distributiva del producto respecto de la suma dice que $a \cdot (b + c) = a \cdot b + a \cdot c$. Si expresamos estas dos propiedades de los números naturales como si fueran afirmaciones, llamando a la propiedad asociativa proposición H y a la distributiva proposición I, sustituyendo además a, b y c por sus valores:

$$H = [(2 + 3) + 5 = 2 + (3 + 5)] \Rightarrow [H \text{ es } \ldots],$$

$$I = [2 \cdot (3 + 5) = 2 \cdot 3 + 2 \cdot 5] \Rightarrow [I \text{ es } \ldots],$$

tendremos que no hay programa de ordenador ni máquina alguna que pueda, de un modo automático, demostrar o refutar la veracidad de la totalidad de este tipo de afirmaciones. Aunque resulte frustrante no se puede escribir un programa de ordenador que demuestre algo tan evidente para nuestra intuición, incluso para un niño en edad escolar, como es $(2 + 3) + 5 = 2 + (3 + 5)$. Por tanto, hay en la matemática «proposiciones verdaderas» acerca de los números cuya veracidad no puede ser probada por medio de la aplicación de reglas de deducción. Como es fácil de imaginar el teorema de Gödel hizo tambalear la aparente solidez de las ideas de Bertrand Russell, y lo que es peor, los mismos pilares del edificio formal de la matemática del que los matemáticos se sienten tan orgullosos.

Uno de los matemáticos más influyentes del siglo XIX y principios del XX, el alemán David Hilbert, dijo que toda esta discusión podía reducirse a un problema de determinación, esto es, de poder

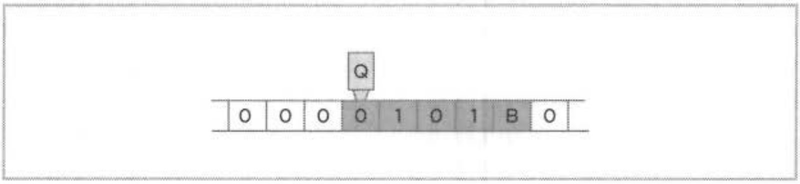
establecer la consistencia o inconsistencia de un sistema formal. Esto significa que hasta la fecha los matemáticos «hacían su ciencia» usando reglas de deducción —es decir, razonando— y axiomas, esto es, ideas o proposiciones que se consideran evidentes y por ello no requieren demostración alguna. Fue en este contexto cuando Hilbert propuso a la comunidad científica el desafío de encontrar un procedimiento mecánico, o en lenguaje actual un «procedimiento de computación», con el que fuera posible decidir la veracidad o no de una proposición matemática. Se trataba de abandonar la discusión meramente académica iniciada por Gödel y buscar una solución real a este problema, ya que nada más y nada menos era la «honorabilidad» de la matemática la que estaba en juego. Alan Turing no pudo resistirse a tal desafío, como era propio de su personalidad, así que se puso a trabajar en busca de la solución al problema lanzado por Hilbert, consecuencia a su vez del teorema enunciado por Gödel: esta consistió en una máquina teórica, sin existencia real, a la que Turing bautizó originalmente como *máquina-a*. Este dispositivo, conocido popularmente como *máquina de Turing*, nació de una discusión al más alto nivel entre filósofos y matemáticos. En la actualidad se considera que es la propuesta a nivel teórico del primer ordenador de la historia de la ciencia. Sin embargo, pese a la genialidad de las ideas que Turing manejaba en 1937, estas no eran suficientes para que se materializaran realmente en la construcción de un ordenador. Lamentablemente, hizo falta un conflicto bélico a gran escala, la Segunda Guerra Mundial, para que matemáticos e ingenieros aunaran esfuerzos para conseguir que se diseñara y construyera esa máquina asombrosa: el ordenador.

Ahora bien, ¿qué es realmente una máquina de Turing, qué partes o dispositivos tiene? Una máquina-a —abreviatura de máquina-a(utomática)— es un dispositivo abstracto, sin existencia real, que representa la configuración más sencilla de un ordenador. La máquina es tal que es capaz de leer y escribir símbolos sobre una cinta dividida en celdas, que es teóricamente infinita. Esto significa que no tiene fin por su derecha ni por su izquierda. Obviamente la cinta representa la memoria principal; en un ordenador actual el equivalente sería la memoria RAM. Es interesante observar que Turing

definió una memoria ilimitada, anticipándose y dando la relevancia a uno de los elementos más importantes en el origen e historia de los ordenadores: la memoria. Más aún, por razones evidentes los ordenadores no pueden disponer de una memoria de tamaño ilimitado, lo que explica que estos se «cuelguen» cuando su memoria no es suficiente al ejecutar un cierto programa o proceso.

Pero ¿qué se graba en la cinta? Supóngase que disponemos de un alfabeto formado solamente por dos dígitos, el 0 y el 1, y un tercer símbolo consistente en «no escribir símbolo alguno», al que llamaremos *blanco* o *B*. El conjunto de estos tres símbolos forma un alfabeto al que llamaremos *A*. Así pues, cada celda de la cinta infinita tendrá inicialmente grabado un símbolo, ya sea 0, 1 o B (véase la figura).

Consideremos ahora una máquina-a en su configuración más elemental: ¿de qué partes se compone? Por un lado, dispone de una cabeza de lectura y escritura, con la cual lee el contenido de una celda, lo borra y graba en su lugar un nuevo símbolo. En el modelo general de máquina de Turing se considera que cada vez que la cabeza de la máquina ha concluido el ciclo de lectura de una celda, borrado su contenido y grabado un nuevo símbolo, la cabeza, y con ella toda la máquina, se mueve una posición hacia la derecha de la cinta (D) o a su izquierda (I). Efectivamente, se puede considerar, siendo equivalentes, que sea la cinta o la máquina, una de las dos, la que dé el salto a D o a I. Y por otro lado, la máquina dispone de una pequeña memoria, el registro, en el que se almacena en qué «estado» o configuración se encuentra en un cierto instante de tiempo de forma similar a, por ejemplo, un semáforo que puede estar en estado rojo (R), ámbar (A) o verde (V). En un instante dado, la máquina se encontrará en un determinado estado, siendo finito el conjunto de posibles estados. A este con-



junto de estados lo representaremos con la letra Q (véase la figura). Supongamos que nuestra máquina del ejemplo puede encontrarse en uno de los siguientes cuatro estados: E1, E2, E3 o E4. Consideraremos también que hay un estado especial, o estado inicial I_0, que es el valor que tiene el registro cuando la máquina es puesta en funcionamiento.

Así pues, la máquina dispone de dos conjuntos finitos de símbolos, los valores que se graban en las celdas de la cinta $A = \{0, 1, B\}$ y los estados del registro de la máquina $Q = \{I_0, E1, E2, E3, E4\}$. Ahora bien, para que la máquina de Turing resulte útil, y por tanto «pueda realizar su trabajo», debe seguir un protocolo similar al de un oficinista. Cada vez que un oficinista realiza un trabajo administrativo su ejecución tiene lugar por pasos sucesivos, tal que concluido un paso debe conocer cuál es el siguiente que debe realizar. De forma similar, cada vez que la máquina de Turing ha procesado un símbolo de la cinta, debe actualizar su estado antes de procesar el símbolo siguiente.

Para que la máquina de Turing pueda cambiar de estado se define una tabla, la denominada *tabla de acciones*, que identificaremos con el símbolo Δ. La tabla, conocida también como *función* o *reglas de transición*, indicará a la máquina qué estado u operación futura deberá efectuar una vez concluida la operación anterior. Por tanto, gracias a la lectura de dicha tabla la máquina de Turing actualizará su estado, una vez concluida la tarea actual. Cada vez que la cabeza de lectura/escritura lee un símbolo de la cinta, lo «combina»

LOS ESTADOS DE UNA MÁQUINA

Un ejemplo simple y cotidiano de los posibles estados para una máquina son los programas de lavado de una lavadora. Cada vez que la máquina ha concluido una cierta tarea, debe actualizar su estado, siguiendo el programa que le hayamos marcado, normalmente el programa de lavado estándar, con prelavado, lavado, aclarado y centrifugado. Es decir, en este caso, los estados de la máquina (la lavadora) son las diferentes partes del programa de lavado que puede estar ejecutando en un momento determinado.

con el símbolo que representa su propio estado en la tabla, en la que está grabado qué «deberá hacer» la máquina a continuación para cada una de las combinaciones de símbolos. Es decir, en la tabla se representa el estado de celda en la cinta y el estado de la máquina, esto es $A \times Q$. La situación futura de la máquina queda definida en la tabla a partir de la siguiente información: cuál es el estado futuro Q de la máquina, y cuál será el nuevo símbolo A que deberá escribirse en la cinta en sustitución del símbolo leído, así como en qué sentido tendrá que desplazarse, si hacia la derecha (D) o hacia la izquierda (I). Por tanto, en su forma más sencilla, una máquina de Turing está definida por tres elementos: los estados de la máquina Q, un alfabeto de símbolos A que se escriben y borran en una cinta de memoria, y una tabla Δ que recogerá para cada paso concluido cuál es el paso siguiente que deberá realizar la máquina de Turing.

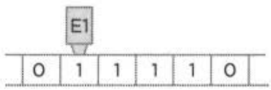
Con el fin de entender el funcionamiento de la máquina de Turing, supóngase un ejemplo elemental con tres estados $Q = \{E1, E2, E3\}$ y una cinta de memoria cuyas celdas pueden contener los símbolos $A = \{0, 1\}$. Asumamos que hemos asignado su estado inicial I_0 igual a E1 y que la cabeza de lectura/escritura está sobre la segunda celda a la izquierda del fragmento de la cinta que estudiaremos, en el ejemplo 011110. Sea la tabla de acciones la formada por las tres tablas menores, una para cada estado de la máquina E1, E2 o E3, que se muestran más abajo; ¿cuál es el comportamiento que exhibirá la máquina?

E1			
Símbolo cinta	Escribe símbolo cinta	Mover	Próximo estado máquina
0	1	I	E2
1	0	D	E3

E2			
Símbolo cinta	Escribe símbolo cinta	Mover	Próximo estado máquina
0	0	I	E3
1	1	D	E1

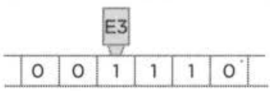
E3			
Símbolo cinta	Escribe símbolo cinta	Mover	Próximo estado máquina
0	1	I	E1
1	0	D	E2

Leyendo en la tabla de estados, y asumiendo que realiza una operación en cada unidad de tiempo (t_o, t_1, t_2...), tendremos en el estado inicial, t_o:



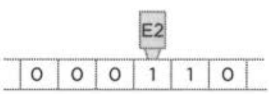
t_0

De acuerdo con la tabla de los estados de la máquina, y puesto que la máquina en el tiempo inicial t_0 está en estado E1 y el símbolo en la cinta es 1, entonces escribirá 0 en la celda y se moverá una celda hacia la derecha, actualizando su estado a E3.



t_1

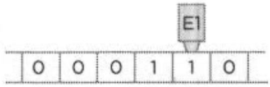
A continuación, para el siguiente instante de tiempo, t_1, al estar la máquina en estado E3 su comportamiento será aquel que está especificado en la tabla de estados. Por consiguiente, puesto que en la cinta el símbolo que lee la cabeza de lectura/escritura es 1, adoptará el estado E2, escribirá un 0 en la celda, y se desplazará de nuevo una celda a la derecha.



t_2

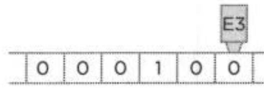
Una vez concluida la tarea anterior, el tiempo se incrementará una unidad, encontrándose ahora la máquina en el instante t_2. Puesto que la máquina se halla en estado E2 y el símbolo de la celda de la cinta de memoria que lee la cabeza de lectura/escritura es 1, entonces, de nuevo obedeciendo las indicaciones de la tabla

de estados, escribirá un 1 en la celda, se desplazará una vez más hacia la derecha y actualizará su estado a E1.



t_3

Concluiremos el ejemplo actualizando el estado de la máquina de Turing para el instante t_4. Considerando que la máquina está en estado E1 y la celda que está leyendo está en 1, entonces escribirá un 0 en la celda, se desplazará una celda a la derecha y adoptará el estado E3.



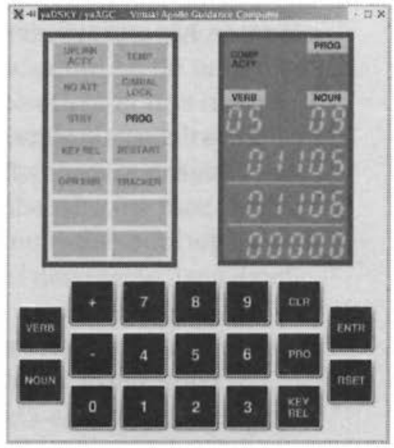
t_4

LA «MÁQUINA-U» DE TURING: ¿PUEDE UNA MÁQUINA SER UNIVERSAL?

Una de las limitaciones de la máquina de Turing es que se comporta como un ordenador que tuviese siempre un mismo programa, y por tanto únicamente podría realizar una sola tarea. Desde un punto de vista histórico uno de los primeros ejemplos de máquina de Turing fue el sistema AGC (Apollo Guidance Computer). Esta máquina fue el ordenador principal de a bordo de las misiones Apolo de la NASA que permitieron la hazaña de llevar al hombre a la Luna el 20 de julio de 1969. Mucho antes de esta epopeya, y consciente de esta limitación, Alan Turing introdujo una generalización de su máquina, a la que se denominó *máquina de Turing universal* o *máquina-u*. Se trata de una máquina de Turing que es capaz de simular cualquier otra máquina de Turing, y por tanto capaz de procesar distintos programas. Por consiguiente, un ordenador es un ejemplo de máquina de Turing universal. Otro ejemplo son los *Smartphones*, teléfonos móviles con prestaciones de un miniordenador.

LA MISIÓN APOLO 11 A LA LUNA

Uno de los ejemplos de máquina de Turing más interesantes es el miniordenador de las misiones Apolo, organizadas por la NASA con la finalidad de llevar al hombre a la Luna. Se trataba de una máquina de Turing desarrollada por el Instituto Tecnológico de Massachusetts, que solo servía para la navegación y el alunizaje. De entre los muchos miniordenadores construidos para diferentes misiones, el AGC (Apollo Guidance Computer) fue uno de los más populares. Por otro lado, el Virtual AGC es un programa con el que es posible simular el miniordenador de las misiones Apolo y con el que incluso es posible ejecutar programas originales en un ordenador actual, ya sea bajo Windows, Linux, Mac OS X u otros sistemas operativos. Su programación es en lenguaje ensamblador, un lenguaje de programación de bajo nivel, dada la capacidad de memoria del microprocesador del AGC, de 38912 palabras en una longitud de 15 bits (una secuencia de quince unos y ceros). El funcionamiento consiste en simular un ordenador virtual dentro de la máquina AGC, que hacía la función de intérprete, es decir, ejecutaba el programa almacenado en la memoria. En el miniordenador AGC del módulo lunar se utilizó el programa Luminary, mientras que en el AGC del módulo de mando se usó el programa Colossus; ambos programas están disponibles en el simulador.



El miniordenador de las misiones Apolo simulado en el emulador Virtual AGC.

El hecho de que una máquina de Turing pueda ser universal representa un paso decisivo en la historia de los ordenadores. Si junto a este hecho consideramos también otro de suma importancia, la conocida tesis de Church-Turing, concluiremos que la invención del ordenador estaba ya próxima. El matemático estadounidense Alonzo Church, una de las figuras más importantes en lógica matemática, formuló con Alan Turing lo que se ha bauti-

zado como *tesis de Church-Turing*. En un lenguaje actual, su tesis establece que la clase de problemas que puede resolver una máquina de Turing universal, y por tanto un ordenador, son los que su solución pueda ser expresada por medio de un algoritmo. No obstante, hay que tener en cuenta que en aquella época el vocablo *algoritmo* no se utilizaba aún, y para referirse a este concepto lo hacían con la expresión «método efectivo de computación». Podemos definir un algoritmo como el conjunto de pasos o reglas que conducen al resultado o solución de un problema. Por consiguiente, para un ordenador un algoritmo es sinónimo de solución. Todo algoritmo debe cumplir ciertas propiedades:

— En primer lugar, el número de pasos que conducen a la solución ha de ser finito, es decir, el protocolo que se recorre hasta la solución debe concluir siempre, por largo que sea.

— En segundo lugar, los pasos o reglas deben estar bien definidos, sin ambigüedades. Para ilustrar esta idea, considérese, por ejemplo, un sencillo experimento escolar consistente en «medir el número π»: primero, rodearemos una lata cualquiera con una cinta de papel, cortando el material sobrante de la cinta; segundo, retiramos la cinta de papel y medimos con una regla su longitud; tercero, situamos la lata entre dos libros, midiendo la distancia entre los bordes de los libros en contacto con la lata para obtener su diámetro, y cuarto, calculamos el cociente entre la longitud y el diámetro, el valor obtenido es el valor de π.

— En tercer lugar, aunque este es un requisito opcional, lo ideal será que un algoritmo pueda resolver no un problema concreto, sino problemas de una misma clase, por ejemplo, ordenar palabras alfabéticamente.

— Y en cuarto lugar, también requisito opcional, que el camino hasta la solución conste del menor número posible de pasos.

Por ejemplo, el protocolo para poner en marcha una lavadora responde al siguiente algoritmo:

— Paso 1. Clasificar la ropa según su color. Se lavarán por separado las prendas blancas y de colores claros de las prendas de color u oscuras.

— Paso 2. Leer en las prendas su etiqueta para averiguar la temperatura máxima y tipo de lavado (así como secado, planchado y otros).

— Paso 3. Introducir en la lavadora el detergente y suavizante.

— Paso 4. Seleccionar el programa y la temperatura idónea.

— Paso 5. Pulsar el botón de puesta en marcha de la lavadora.

— Paso 6. Sacar la ropa.

— Paso 7. Fin.

Las matemáticas que se estudian en edad escolar están repletas de sencillos algoritmos. Por ejemplo, la resolución de sistemas de ecuaciones por el método de sustitución consiste en el siguiente algoritmo:

— Paso 1. Se despeja la misma incógnita en ambas expresiones.

— Paso 2. Igualar las expresiones.

— Paso 3. Resolver la ecuación.

— Paso 4. Sustituir el valor obtenido en cualquiera de las expresiones en las que estuviera despejada la otra variable.

— Paso 5. Resolver la ecuación resultante en el paso anterior.

— Paso 6. Fin.

De estas consideraciones concluiremos que un ordenador es una máquina de Turing universal que procesa algoritmos. Cuando

ALONZO CHURCH, EL CÁLCULO LAMBDA Y LISP



Pese a que la figura de Alan Turing ha estado siempre ligada a la máquina que lleva su nombre, lo cierto es que cuando dio a conocer su trabajo otro matemático de gran talla, Alonzo Church (1903-1995), había publicado un trabajo con el que restaba algo de originalidad al realizado por Turing. Durante los años treinta, Church introdujo, junto con Stephen Kleene (1909-1994), el denominado *cálculo* λ, una abstracción matemática con la que estudiar qué es una función. Una función es una expresión matemática $y=f(x)$ que relaciona el valor de dos variables, por ejemplo la longitud x y el peso y en las ballenas azules por medio de la expresión $y=3{,}15\,x-192$. Aunque este concepto fue introducido en el siglo XVII por Descartes, Newton y Leibniz, fue de nuevo reconsiderado en los años treinta del siglo XX con el fin de elaborar una teoría general sobre funciones matemáticas.

Una nueva sintaxis

Uno de los méritos de Church fue introducir una nueva sintaxis con la que representar a esta clase de expresiones matemáticas. Así, por ejemplo, si evaluamos la expresión (+(*2 3)(*5 6)) —el asterisco es el operador multiplicación—, entonces tendremos que el resultado es 36, ya que $(2\cdot 3)+(5\cdot 6)=$ $=6+30=36$. Por consiguiente, una función matemática sería una abstracción. De una forma similar, para el cálculo λ, se usa una expresión algo más enrevesada, (λ*x*. +*x* 1), que significaría lo que sigue: «La función (representada por

la solución a un problema es expresable por medio de un algoritmo, se dice entonces que el problema es *computable*. El ingeniero suizo Niklaus Wirth (n. 1934), autor de los lenguajes de programación Algol, Modula-2 y Pascal, entre otros, introdujo en 1975 la definición de un programa. De acuerdo con su definición, el *programa* es la reunión del algoritmo con la forma de organizar los datos dentro del

el símbolo λ) de la variable (aquí *x*) —concepto que de una forma convencional escribiríamos como λ (*x*)— que (representado como .) añade (por el operador +) el valor de la variable (es decir, *x*) a 1». Podemos complicar un poco más la expresión anterior, por ejemplo escribiendo ((λ *x*.+*x*1) 3), cuyo resultado sería 4, porque estamos diciendo que el valor de *x* es 3. Como es previsible podemos ir complicando las operaciones hasta desarrollar todos los elementos del cálculo λ. Otro de los méritos de esta clase de cálculo fue que tuvo un profundo impacto sobre la teoría que estudia la programación de ordenadores.

El problema de la parada

Pero si por algo es célebre el cálculo λ es porque Church utilizó este formalismo para estudiar el llamado *problema de la parada,* obteniendo como resultado la noción de *problema computable*, que es precisamente la idea que subyace en la máquina de Turing. A su vez, Turing demostró en 1937 que tanto el cálculo λ como su máquina eran equivalentes, es decir, permitían llegar por dos vías diferentes a los mismos resultados. Cuando una máquina de Turing procesa alguna de las expresiones indicadas, por ejemplo (+3 1), se detiene una vez obtenido un resultado, 4 en el ejemplo, siendo esta la expresión computable. Más aún, y desde un punto de vista práctico, el cálculo λ inspiró el desarrollo de los llamados *lenguajes de programación funcionales,* uno de cuyos ejemplos es Lisp, uno de los lenguajes más importantes en inteligencia artificial. Este lenguaje fue introducido en 1958 por John McCarthy (1927-2011), uno de los padres de la inteligencia artificial, y entre cuyas características heredadas del cálculo λ se encuentra el uso de paréntesis:

```
(defstruct persona
   (nombre 'Alan)
   (edad 41))
```

o en el caso más sencillo:

```
(format t «¡Hola Turing!»)
```

programa, lo que se conoce como *estructura de datos*, proponiendo una de las expresiones más célebres heredera del trabajo de Turing: *algoritmo + estructura de datos = programa.*

OTRAS MÁQUINAS DE TURING

En 1982, el premio Nobel de Física Richard Feynman (1918-1988) planteó una cuestión realmente apasionante y que volverá a ser tratada en el último capítulo. Predijo la clase de problemas que no podrían ser tratados jamás con un ordenador, tras encontrar una limitación en la capacidad computacional de las máquinas de Turing, además del denominado *problema de la parada*, que trataremos en el siguiente apartado. Feynman propuso que tanto las máquinas de Turing como los ordenadores en general no podían ser aplicados a la simulación de fenómenos de naturaleza cuántica, es decir, los que se observan en los átomos y para los que la física clásica es insuficiente. Con esto quería decir que un fenómeno cuántico es *no computable* y, por tanto, no podía ser tratado con un ordenador convencional. Para que esto fuera posible, según Feynman, una máquina de Turing tendría que ser capaz, entre otras peculiaridades, de estar en varios estados simultáneamente o leer al mismo tiempo varias celdas de la cinta. Extrapolando estas características a un ordenador actual, el ordenador en cuestión tendría que poder manipular, además de los estados 0 o 1, posibles «estados intermedios» entre 0 y 1, y leer «a la vez» varios registros de la memoria RAM. No obstante, una vez propuesto el límite en la aplicación de la máquina de Turing, otro físico, el anglo-israelí David Deutsch (n. 1953), introdujo en 1985 una nueva clase de máquina de Turing con la que esta limitación quedaría definitivamente superada, la máquina de Turing cuántica. Los ordenadores cuánticos podrían simular problemas no computables, como son los fenómenos cuánticos, y, obviamente, tendrían numerosas aplicaciones en el mundo real.

Además de la máquina original introducida por Turing y de su versión cuántica, otros diseños han sido propuestos. Por ejemplo, es posible construir una máquina de Turing policefálica, es decir,



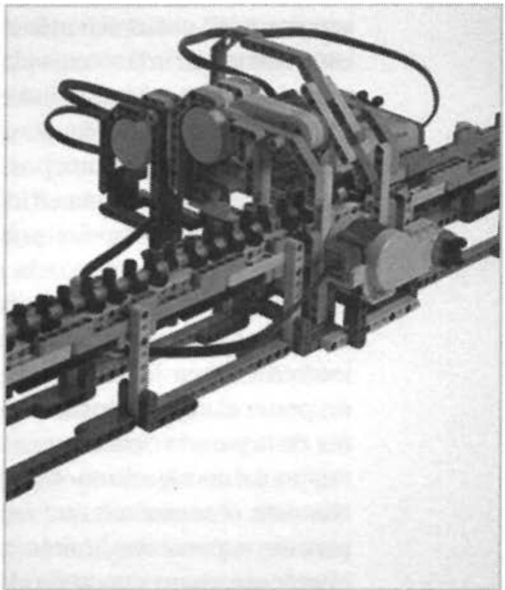




FOTO SUPERIOR IZQUIERDA:
Kurt Gödel (1906-1978), padre del teorema de incompletitud, que hizo tambalearse los cimientos de la matemática.

FOTO SUPERIOR DERECHA:
Detalle de una máquina de Turing construida con piezas de LEGO.

FOTO INFERIOR:
Alan Turing participando en una carrera de fondo en Dorking, Inglaterra, en 1946, en la que quedó en segunda posición.

una máquina con dos o más cabezas de lectura/escritura que leen y escriben sobre una misma cinta, lo que aumenta su eficiencia computacional. Otra posibilidad es la máquina de Turing capaz de leer datos en celdas de más de una cinta. También se han propuesto otras alternativas, como, por ejemplo, la máquina de Turing no determinista, una máquina en la que la tabla de acciones contiene más de una regla de transición para un cierto estado, eligiéndose al azar la regla de transición con la que se actualizará su estado. Sin embargo, el diseño que representó un verdadero desafío es la clase de máquina a la que Turing denominó *oráculo* o *máquina-o*. Con ella intentó superar los límites de su máquina convencional, dotándola de poder computacional suficiente como para resolver el problema de la parada o problemas cuya solución no fuera expresable por medio de un algoritmo. Una máquina-o es una máquina de Turing que está conectada a una caja negra, denominada *oráculo*, que le permite superar sus limitaciones. Si se prefiere, puede pensarse en el oráculo como una segunda cinta en la máquina de Turing. Para consultarla, esta utiliza un símbolo especial llamado *marcador*. Entre dos marcadores se sitúa el símbolo sobre el que la máquina quiere consultar al oráculo. Seguidamente, la máquina de Turing pasa a un estado especial denominado *estado llamada*, enviando así una petición al oráculo. Si este reconoce el símbolo como perteneciente a su conjunto de símbolos, entonces la máquina pasará al estado-1 y, en caso contrario, es decir, si el oráculo no reconoce el símbolo en cuestión, pasará al estado-0. La máquina-o fue un primer intento realizado por Turing de lo que con posterioridad se ha llamado *hipercomputación*, propuestas que van más allá de la idea de computación introducida por el propio científico inglés.

EL PROBLEMA DE LA PARADA: ¿POR QUÉ SE CUELGA UN ORDENADOR?

Una vez ideada la máquina de Turing, el científico inglés estudiaría un «problema de decisión» por medio de su propia invención, conocido desde entonces como *problema de la parada* (*halting pro-*

blem en inglés o *Entscheidungsproblem* en alemán). El problema consiste en predecir si cuando una máquina de Turing lea un símbolo de la cinta continuará funcionando o por el contrario se detendrá, «colgándose», de forma similar a como lo hacen los ordenadores actuales. Por consiguiente, la pregunta que intentaba resolver Turing no es otra que la posibilidad de que exista un procedimiento mecánico, en la actualidad un programa de ordenador, con el que sea posible establecer si otro programa se detendrá cuando reciba como entrada un cierto valor o *input*. Hoy en día, con cualquier ordenador de usuario es fácil hacer algunos experimentos sencillos sobre estas y otras cuestiones teóricas planteadas por Turing. Si asumimos una equivalencia entre la máquina de Turing y un ordenador en el que ejecutemos un programa, el problema consistirá en decidir si dicho programa detendrá su ejecución o por el contrario se ejecutará indefinidamente. Experimentaremos estas dos situaciones con los siguientes programas en lenguaje BASIC-256. Por ejemplo, el siguiente programa se detendrá apenas se ejecute una vez:

```
print "Hola Turing!"
```

mientras que este otro programa se ejecutará, una y otra vez, sin detenerse nunca:

```
r=true
while r
  print "Hola Turing!"
end while
```

Sin embargo, el problema estudiado por Turing y sus contemporáneos no es tan sencillo como aquí lo presentamos, ya que no se puede desarrollar un procedimiento general que sea capaz de llegar a conclusión alguna sobre la ejecución o parada de un programa cualquiera. El reto consiste en escribir un programa que pueda tomar una decisión sobre esta cuestión, una vez que recibe como datos de entrada o *input* no unos números, por ejemplo el PIN de una tarjeta de crédito, o palabras, por ejemplo un nombre

y los apellidos, sino otro programa. Concluiremos en este capítulo que el problema de la parada es indecidible con una máquina de Turing, pero ¿y con un ordenador?

Supóngase que llamamos `parada(candidato, entrada)` a un programa que es capaz de establecer si otro programa, al que llamaremos `candidato`, detendrá o no su ejecución verificándose su parada o *halt* cuando recibe un cierto valor de entrada o *input*, cuyo valor denominaremos `entrada`. Efectivamente, si representamos `parada(candidato, entrada)` en forma de pseudocódigo, tendremos que:

```
programa parada(candidato, entrada)
  if input=entrada y candidato → se detiene
    then parada(candidato, entrada)=verdadero;
  if input=entrada y candidato → no se detiene
    then parada(candidato, entrada)=falso;
```

Supóngase que utilizando el programa `parada(candidato, entrada)` escribiéramos un nuevo programa, al que denominaremos `paradoja(entrada)`:

```
programa paradoja(entrada)
  if parada(entrada, entrada)=falso
    then return verdadero
    else return falso
```

Demos un paso más en el razonamiento, tal y como hizo Alan Turing, y llamemos `P` al programa paradoja. A continuación, ejecutemos `parada(P, P)`. Si el programa que está dentro del principal devuelve falso, es decir, el programa `P` no se detiene al recibir como valor de entrada o *input* un programa idéntico a él, entonces el programa principal `paradoja(P)` devolverá verdadero, deteniéndose su ejecución, lo que no es cierto y por tanto es «mentira».

Por el contrario, si `parada(P, P)` devuelve verdadero, puesto que el programa `P` detiene su ejecución al recibir un valor similar de entrada `P`, entonces `paradoja(P)` no detiene

su ejecución, siendo también «mentira». Por consiguiente, Turing concluyó que dadas estas contradicciones el programa `parada`, o *halt* en su versión original, carece de utilidad como procedimiento que permita la evaluación de `P`. En otras palabras, el problema de la parada o *halting problem* es un problema irresoluble.

No obstante, y aunque no exista ningún programa que sirva de herramienta universal para resolver satisfactoriamente el problema de la parada, sea cual sea el programa `P`, los científicos pensaron que tal vez resultase factible escribir un programa que devolviera únicamente respuestas a casos, es decir, en lenguaje actual, programas particulares. Esta clase de programas fue bautizada con el nombre de *programas PHS (partial halting solver)* o *solucionadores parciales de la parada*. Sin embargo, tiempo después se concluyó que la situación era tan intratable como el problema de la parada. Por ejemplo, utilizando una vez más el lenguaje BASIC-256, escribamos un programa que recibe como entrada o *input* un programa `P$`. Su tarea consiste en proporcionar como salida o *output* un comentario informando si el programa `P$` detiene o no su ejecución:

```
input P$
  if P$ = "halt" then
    print "el programa SÍ se detiene"
  else
    print "el programa NO se detiene"
  endif
end
```

De acuerdo con los razonamientos anteriores, la conclusión a la que llegamos es realmente decepcionante, ya que no podemos asegurar que este programa de apariencia tan sencilla proporcione al usuario únicamente respuestas correctas. Asombrosamente, antes de que los ordenadores, y, por tanto, el *software*, existieran, Turing fue capaz de llegar a la siguiente conclusión: no existe ningún procedimiento mecánico, ya sea una máquina de Turing o, en lenguaje actual, un programa de ordenador, con el

que se pueda determinar si otro programa (o máquina de Turing) se detendrá *(halt)* dado un cierto valor de entrada o *input*. Esta conclusión fue obtenida por Turing mediante su propio objeto de

UN SINFÍN DE MÁQUINAS DE TURING

Un ordenador actual podría ser considerado como una máquina de Turing que a su vez contiene a otra en su interior. Con el fin de explicar esta idea considérese por ejemplo ENIAC (Electronic Numerical Integrator And Computer), uno de los primeros ordenadores. Este mastodonte de los primeros años de la informática podría ser representado como una máquina de Turing con tres cintas: una cinta de lectura que recibiría los datos de entrada o *input*, otra de escritura que mostraría los resultados, salida o *output*, y una tercera que desempeñaría el papel de una memoria.

Los ordenadores actuales

En un ordenador actual, la máquina de Turing que representa ENIAC debería ser modificada actualizándola, considerando ahora que la cinta de entrada se desdoblaría en dos segmentos: el primero sería la memoria auxiliar, por ejemplo el disco duro, una tarjeta de memoria SD o un *pendrive*, mientras que el segmento restante sería el teclado. En dicha máquina la cinta de salida estaría representada por el monitor, mientras que la de memoria sería la memoria RAM. Si ahora asumimos que el sistema operativo es también una máquina de Turing, por ejemplo, las distintas versiones de Windows de Microsoft, o cualquiera de las distribuciones de Linux/Unix o la familia Mac OS de los ordenadores Apple, entonces tendremos que el conjunto de programas que permiten que un usuario gestione los recursos de un ordenador es una máquina de Turing que controla a otra, el ordenador propiamente dicho. Más aún, cuando un programador escribe un programa, el conjunto de sus instrucciones, esto es, el llamado *código fuente*, debe ser traducido a código máquina o binario con un programa llamado *compilador*, que puede ser también considerado otra máquina de Turing. Una vez traducido, el programa ya podrá ser ejecutado en el microprocesador, el dispositivo más importante del ordenador. Por tanto, el modelo que subyace es que tanto el ordenador como el programa con el que traducimos un programa a una versión ejecutable, o el mismo sistema operativo son todos ellos máquinas de Turing. En otras palabras, «todo son programas, todo es *software*», y aquí se incluyen también los circuitos electrónicos, el *hardware*, como si de *software* se tratase, una de las ideas más apasionantes consecuencia del trabajo que tiempo atrás realizase Alan Turing.

invención, la máquina de Turing. Esto demuestra, una vez más, la genialidad de este científico, que pese a su corta existencia fue uno de los más grandes del siglo XX.

CONSTRUIR MÁQUINAS DE TURING

Aunque resulte paradójico, la máquina de Turing jamás fue llevada a la realidad por su autor, pese a sus denodados esfuerzos. Este artificio fue y es una máquina teórica cuyo mérito es que ha permitido definir qué problemas son o no tratables con un ordenador. Curiosamente, investigadores y aficionados a los ordenadores de todo el mundo han construido, con fines recreativos, la máquina que una vez propusiera este genial científico.

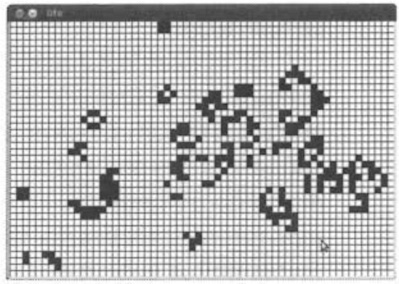
Uno de los primeros modelos fue construido en 1972 en la Universidad de Brandeis, en Massachusetts (Estados Unidos), por Ira Gilbert con el fin de enseñar nociones de programación a los estudiantes. Más recientemente, utilizando el juego de construcciones LEGO, han sido construidas varias versiones de la máquina de Turing. A base de ladrillos de plástico interconectables, Denis Cousineau construyó una máquina de Turing, aunque este modelo no fue enteramente mecánico. Su autor utilizó un ladrillo inteligente de LEGO, llamado *RCX*, de uso habitual en experimentos de robótica recreativa, para almacenar en su microcontrolador programable la tabla con los estados de transición. Otro modelo de máquina de Turing construido con LEGO fue el montado por el japonés Joe Nagata. En 2010 Mike Davey construyó un modelo con aspecto *vintage* en un intento por rememorar la máquina descrita en el trabajo publicado por Alan Turing en 1936. No obstante, su máquina utilizaba un microcontrolador Parallax Propeller y una tarjeta SD en la que se almacenan los estados de la máquina.

De estos experimentos se deduce que la construcción física, o a nivel de *hardware*, de una máquina de Turing no resulta tan sencilla y, de hecho, los modelos de este tipo no abundan. Por el contrario, encontramos numerosos ejemplos de *software* con los que realizar la simulación de dicha máquina, básicamente porque

resulta mucho más sencillo de abordar. Entre los proyectos más interesantes destacan «Turing and Post Machines: C++ Simulators», una colección de programas en lenguaje C++ con los que simular distintas clases de máquina (determinista, no determinista, universal, «con fallos», varias cintas, etc.). El Visual Turing, diseñado para el sistema operativo Windows, es un atractivo simulador con el que es posible experimentar con diversas máquinas de Turing por medio de instrucciones gráficas. Otro de los simuladores es Jflap, esta vez en Java, muy interesante al incluir la simulación de la máquina de Turing junto con otras máquinas similares. Otro ejemplo de máquina de Turing en lenguaje Java, aunque de presentación más sobria, y por tanto de aspecto más académico, es tmsim_bgm. Volviendo la vista atrás, para el sistema operativo MS-DOS, el programa original jkturing, de John Kennedy, de la Universidad de Santa Mónica, en Estados Unidos, representa otra opción de simulación, aunque algo menos atractiva que Visual Turing o Jflap; no obstante, este programa fue actualizado para las distintas versiones del sistema operativo Windows. Muy atractivo es Uber Turing Machine, desarrollado en 2011, que incluye un alfabeto con el que escribir distintos programas para la máquina. Todos estos programas son ejemplos de algo que resulta verdaderamente curioso, pues consisten en simular máquinas de Turing en una máquina de Turing universal, el ordenador.

Uno de los desafíos más interesantes es la posibilidad de construir una máquina de Turing utilizando para tal fin otra máquina, conocida con el nombre de Juego de la Vida. Este juego fue diseñado en 1970 por John Horton Conway (n. 1937), profesor de la Universidad de Cambridge, donde también estudió Turing. Se trata de una simulación por ordenador que fue muy popular entre los aficionados a la ciencia recreativa, especialmente tras su publicación por el matemático y divulgador Martin Gardner (1914-2010) en la revista *Scientific American*. El juego es un autómata celular, es decir, una rejilla en dos dimensiones cuyas celdas están ocupadas por lo que se denominan *autómatas finitos*, también conocidos como *máquinas de estados finitos*; se trata de un objeto que se encuentra en uno de entre un conjunto de estados posibles, siendo este conjunto finito. Por ejemplo, un semáforo

CONSTRUIR LA MÁQUINA DE TURING CON EL JUEGO DE LA VIDA



Captura de pantalla de un momento del desarrollo del Juego de la vida.

A finales del siglo XX una serie de científicos y aficionados a los ordenadores se plantearon la siguiente pregunta: ¿es posible construir una máquina de Turing con el Juego de la Vida? Efectivamente, el 2 de abril de 2000 Paul Rendell consiguió simular una máquina de Turing con el autómata celular creado por John Horton Conway, y volvió a repetir la hazaña el 10 de febrero de 2010. En el primer modelo utilizó una rejilla de 1714x1647 con la que simuló mediante autómatas finitos la máquina-a de Turing. Esta disponía de tres estados posibles y era capaz de procesar tres símbolos distintos en la cinta de memoria. En los experimentos realizados en 2010 simuló una máquina universal o máquina-u de Turing. La posibilidad de simularla mediante el Juego de la Vida condujo a sorprendentes conclusiones: ¿significaba que el Juego de la Vida tenía capacidades similares a un ordenador? Efectivamente, así es. Más aún, cualquier fenómeno de la naturaleza, por ejemplo la formación de anillos de Saturno o la interacción entre conejos y lobos, que pueda ser simulado con un autómata celular, exhibirá también, por sorprendente que parezca, algunas de las características presentes en un ordenador. Otras experiencias similares de «construcción» de máquinas de Turing con el Juego de la Vida han sido realizadas con éxito, y han recibido nombres tan singulares como MRM (Minsky Register Machine), o su versión universal URM, CoreWorld, LogiCell, etcétera.

puede estar en un cierto tiempo t en un estado «verde», de entre tres posibles estados {rojo, ámbar, verde}. Otro ejemplo es una neurona o célula nerviosa, que puede estar en reposo o en estado excitado a consecuencia de la entrada de un estímulo. De forma similar a la máquina de Turing, en un experimento de simulación con un autómata celular, a medida que transcurre el tiempo (t), se irán actualizando los estados de cada autómata finito. Su actualización, es decir, el cálculo de cuál será su estado en el tiempo si-

guiente $(t+1)$, utiliza un conjunto de reglas que se conocen como reglas de transición, que serán las que actualizarán el estado de cada autómata finito teniendo en cuenta tanto su estado actual como los estados de los autómatas vecinos que se encuentran a su alrededor.

En el caso del Juego de la Vida, cada autómata finito tiene 8 vecinos, los situados a su alrededor en posiciones N, S, E y O, así como los de las celdas en la diagonal NE, SE, SO y NO. Además se asume que los autómatas finitos tienen solo dos estados posibles, el estado 0 («muerto») o el estado 1 («vivo»), a los que asignaremos un color arbitrario. El juego actualiza los estados de los autómatas finitos aplicando las siguientes reglas de transición:

— Regla 1: sea el estado de un autómata finito a_{ij}^{t} 0 o 1. Su estado futuro, es decir a_{ij}^{t+1}, será idéntico al estado anterior si el número de vecinos en estado 1 es igual a 2:

$$a_{ij}^{t+1} = a_{ij}^{t} \text{ si suma de vecinos } (a_{ij}^{t}) = 2.$$

— Regla 2: un autómata finito pasará al estado 1 si la suma de sus vecinos en estado 1 es igual a 3. Por consiguiente, en este caso el cambio de estado del autómata solo tendrá lugar si su estado era 0 en el tiempo *t*, permaneciendo en el estado 1 en caso contrario:

$$a_{ij}^{t+1} = 1 \text{ si suma de vecinos } (a_{ij}^{t}) = 3.$$

— Regla 3: esta regla simula el efecto de una vecindad con una alta o baja densidad de autómatas «vivos», esto es, en estado 1. Si el número de autómatas de la vecindad en estado 1 es inferior a 2 (es decir, uno o ninguno) o fuera superior a 3 (cuatro, cinco, seis, siete u ocho), entonces el autómata finito «muere» adoptando el estado 0. En este caso el cambio de estado del autómata solo tiene lugar si su estado era 1 en el tiempo *t*, manteniendo su estado 0 en caso contrario:

$$a_{ij}^{t+1} = 0 \quad \begin{array}{l} \text{si suma de vecinos } (a_{ij}^{t}) < 2 \\ \text{o suma de vecinos } (a_{ij}^{t}) > 3. \end{array}$$

Aplicando de un modo iterativo las reglas de transición a cada uno de los autómatas finitos que componen el autómata celular tendrá lugar su evolución, pudiéndose observar la aparición de patrones o figuras características de este juego. Sus formas y cambios causaron, y siguen haciéndolo, una gran fascinación entre los aficionados a los ordenadores, de manera que se organizaron concursos y otras actividades. Aunque hay una gran oferta de programas con los que experimentar con el Juego de la Vida (Life32, Xlife 2.0, Life 1.05/1.06, ProLife, Mcell, dbLife, etc.), familiarizándose así con el concepto de autómata, uno de los más completos y espectaculares es Golly.

LA AVENTURA AMERICANA

En agosto de 1936 Alan Turing envió para su publicación en los *Proceedings of the London Mathematical Society* un artículo titulado «Sobre los números computables con una aplicación al Entscheidungsproblem», ya mencionado anteriormente, en el que introdujo su célebre máquina. En dicho artículo define también los conceptos de «computable» y «no computable», e incluye algunas ideas fundamentales, no solo para las matemáticas sino para la informática. Casualmente ese mismo año Alonzo Church publicó en la revista *American Journal of Mathematics* un artículo titulado «Un problema irresoluble de teoría elemental de números»; ambos científicos habían llegado a los mismos resultados, aunque por vías diferentes. Mientras Turing razonaba de manera muy original, considerando la clase de operaciones que de «forma mecánica» podría hacer en el mundo real una persona, por ejemplo, un oficinista que repite una tarea una y otra vez, o una máquina que suma dos números, Church razonaba de una forma clásica, dentro del «mundo abstracto» que es propio de las matemáticas. Lamentablemente Turing publicó poco después que Church los resultados, lo que le restó originalidad al tener que hacer referencia al trabajo del matemático estadounidense. Estas dos publicaciones representan las bases teóricas de lo que más tarde sería el ordenador.

Un mes después, en septiembre de 1936, Turing viajo a Estados Unidos. Una vez allí su idea era completar sus estudios de doctorado durante dos años en el Instituto de Estudios Avanzados de la prestigiosa Universidad de Princeton. Bajo la dirección de Alonzo Church, Turing estudió algo que puede resultar curioso, incluso en la actualidad, el uso de la «intuición» en matemáticas. Sin entrar en cuestiones filosóficas, la intuición podría definirse como un producto del sentido común. Se trataría de una forma de anticipación o visión mental que nos ayuda durante un razonamiento a llegar a un cierto resultado o conclusión. Puesto que durante un razonamiento concatenamos hechos de forma lógica, la intuición sería un «ingrediente extra» del que haría uso un matemático con objeto de obtener la solución a un cierto problema.

«El razonamiento matemático puede considerarse más bien esquemáticamente como el ejercicio de una combinación de dos instalaciones, que podemos llamar la intuición y el ingenio.»

— Alan Turing, «Systems of logic based on ordinals».

Turing proponía que tal vez la intuición humana sea posible gracias a etapas que no pueden ser expresadas por un algoritmo. Dichas etapas «no algorítmicas» tendrían lugar durante un razonamiento, ayudando así al entrelazamiento de los hechos que conducen a un resultado o conclusión. Pero no solo en matemáticas existe la intuición. También un médico o un mecánico de coches hacen uso de ella a la hora de dar un diagnóstico.

Durante esta época Turing comenzó interesarse en la posibilidad de construir su máquina, pero no llevó a cabo su propósito. Fue durante su estancia en Estados Unidos cuando nació su interés por el *hardware*, y por tanto, la posibilidad de construir con circuitos electrónicos o componentes electromecánicos lo que hasta entonces no era más que una recreación mental. Una vez más, tal como ocurrió cuando concibió a «nivel lógico» la máquina de Turing, comenzó a pensar sobre la «parte física» en una época en la que todavía no había ordenadores. En su lugar construyó una máquina multiplicadora con relés electromagnéticos,

con la que era posible multiplicar dos números binarios (números representados utilizando únicamente los dígitos 0 y 1).

En 1938 otro de los personajes geniales que concurrieron en esta misma época, el investigador estadounidense de origen húngaro John von Neumann, ofreció a Turing un puesto temporal en la Universidad de Princeton. Sin embargo, este rechazó la oferta, y en el verano de ese mismo año volvió al King's College. Una vez allí comenzó a construir un mecanismo analógico para evaluar la llamada *hipótesis de Riemann.*

En agosto de 1939 Turing recibió la proposición de incorporarse al Bletchley Park en calidad de criptógrafo para descifrar los mensajes interceptados al ejército nazi.

CAPÍTULO 2

Máquinas contra códigos. Turing criptógrafo

La Segunda Guerra Mundial no fue una guerra más. En ella combatieron soldados y civiles y, entre estos últimos, los científicos de uno y otro bando. Por mar y aire el Reino Unido estuvo sometido a un cruel acoso por parte de la Alemania nazi. Los británicos consiguieron vencer a su enemigo, pero para ello tuvieron que reclutar a sus mejores intelectos, entre ellos Alan Turing. La guerra supuso la promoción de los nuevos hallazgos científicos, como la energía nuclear, y de sorprendentes invenciones, como fue el caso del ordenador.

El 3 de septiembre de 1939 comenzó la batalla del Atlántico, que duraría casi hasta los últimos días de la Segunda Guerra Mundial y se convirtió en uno de los teatros de operaciones más espectaculares de la contienda. Durante prácticamente todo el conflicto, los submarinos alemanes, conocidos popularmente como *U-Boot*, atacaron sistemáticamente a la flota mercante británica, poniendo en peligro en más de una ocasión el abastecimiento de las Islas Británicas. Si bien durante la Primera Guerra Mundial la Marina alemana se había enfrentado en numerosas ocasiones a la británica, nada de lo acaecido hasta la fecha guardaba parecido con el escenario que ahora se desarrollaba. La aparición en escena de una nueva clase de navío sumergible, el submarino, representó para los británicos un arma mortífera sin precedentes, que les obligó a cambiar sus tácticas de navegación, formando convoyes que eran custodiados por buques escolta para una mejor defensa. Esta estrategia ayudó temporalmente a los británicos, ya que los primeros submarinos alemanes eran lentos y tenían que salir a la superficie para disparar, por lo que eran presa fácil de sus enemigos. Al finalizar la guerra, Alemania había perdido un 75% aproximadamente de sus submarinos, con la consiguiente pérdida de vidas que ello conllevó. Por su parte, el acoso alemán causó al Reino Unido serios problemas de abastecimiento, pues el bloqueo no se limitó solo al mar. Entre 1940 y 1941 el país sufrió los terribles «bombardeos relámpago» de

LA BATALLA DEL ATLÁNTICO: LA SIMULACIÓN DE UN «U-BOOT»



El juego de estrategia *Action in the North Atlantic*, un clásico de la simulación de submarinos.

Lejos de lo que Turing y otros contemporáneos suyos hubieran imaginado, una de las aplicaciones con más éxito de los ordenadores es el entretenimiento. La simulación, la imitación de un sistema real, como puede ser la navegación de un submarino, es a día de hoy una de las principales aplicaciones de los ordenadores y permite experimentar situaciones que de otra forma serían inalcanzables para la mayor parte de las personas. Los videojuegos de simulación permiten al jugador aprender el funcionamiento de un sistema (por ejemplo, de navegación), gestionar recursos (como combustible, personal, etc.) o resolver situaciones complejas (por ejemplo, una batalla naval). Los simuladores de submarinos, o «subsim», son un tipo de videojuegos que permiten al jugador dirigir un submarino. El juego consiste por lo general en llevar a cabo una serie de misiones en las que hay que hundir uno o más barcos y sobrevivir al contraataque de los destructores, utilizando mapas, el radar, un periscopio y torpedos.

la Luftwaffe, y aunque Londres fue la principal ciudad bombardeada, otras muchas también fueron atacadas, causando numerosas muertes y la destrucción de cerca de un millón de viviendas.

El 23 de febrero de 1918, tras finalizar la Primera Guerra Mundial, un ingeniero alemán llamado Arthur Scherbius (1878-1929) patentó Enigma, una máquina para cifrar mensajes. La máquina fue comercializada por la empresa Scherbius & Ritter, fundada por su inventor y un socio, aunque posteriormente vendió los derechos de explotación a la también empresa alemana Chiffriermaschinen Aktien-Gesellschaft. A principios de los años veinte, Enigma fue presentada al público en dos ciudades europeas. A partir de ese momento, se comercializó toda una gama de

modelos para uso civil, que llevaron por nombre una sola letra (A, B, C, D). Aunque en un principio fue concebida para cifrar transacciones comerciales, el gran negocio vendría con la guerra. En España se comercializó el modelo D, más tarde utilizado durante la Guerra Civil española. Sin embargo, el mejor cliente fue Alemania, que encargó el desarrollo del modelo G para su Ejército: el modelo Funkschlüssel o M para la Marina y el modelo Wehrmacht o I, uno de los más populares, que será precisamente el que elegiremos como referencia para explicar su funcionamiento. En 1942 los *U-Boot* adoptaron también su propio modelo. Como dato anecdótico cabe decir que el 40% de las máquinas Enigma se fabricaron durante la Segunda Guerra Mundial. De hecho, para los alemanes fue una máquina tan vital, que Hitler ordenó que su fabricación formase parte del programa de armamento del Gobierno del III Reich.

LA MÁQUINA DIABÓLICA. ¿CÓMO FUNCIONABA ENIGMA?

Aunque su aspecto y facilidad de manejo recordaban al de una máquina de escribir, Enigma escondía una complejidad sin precedentes. Su funcionamiento era el resultado del uso de componentes mecánicos y eléctricos. Un teclado y un conjunto de discos o tambores, denominados *rotores*, representaban la parte mecánica de la máquina. Cada rotor tenía representado el alfabeto de veintiséis letras, desde la A hasta la Z. Cuando un operador pulsaba una tecla, se producía el giro de un rotor, a continuación del siguiente y después de uno tras otro de los rotores vecinos, paso a paso, y de manera ordenada de derecha a izquierda. Esta orquestación en el giro era controlada por una hendidura en los rotores, la cual conseguía, mediante este giro acompasado de los rotores, que una misma letra, por ejemplo la A, no fuera codificada siempre por el mismo carácter. Los rotores estaban diseñados de tal forma que cada una de sus dos caras disponía de contactos que formaban un circuito eléctrico con los contactos del rotor vecino. En el interior

de cada rotor había a su vez veintiséis cables que conectaban cada uno de los contactos en una de las caras del rotor con algún otro de los contactos situado en su otra cara. Si a esto añadimos que la trama de cables que conectaban los contactos de las dos caras de cada rotor era diferente de un rotor a otro, el resultado era una máquina diabólica. Lo habitual era que una máquina Enigma tuviera tres o cuatro rotores en línea definiendo, para cada pulsación de una tecla, un cierto circuito eléctrico que era distinto del que resultaba tras pulsar la siguiente tecla, ya fuera la misma o distinta. Por este motivo, el cifrado de un carácter era siempre una letra distinta, dependiendo del circuito eléctrico que en ese momento definieran los rotores.

El manejo de Enigma requería del siguiente protocolo. En primer lugar, antes de cifrar o descifrar un mensaje el operador de la máquina debía colocar de derecha a izquierda los rotores en un cierto orden. A continuación, los rotores se giraban hasta alcanzar una cierta posición inicial, que venía representada por una de las veintiséis letras del alfabeto, que era la única visible a través de un visor del tamaño de un carácter. En un principio el orden de los rotores y su posición inicial fueron las dos características de configuración de las que dependía el cifrado y descifrado de mensajes. A estas dos características se le añadió una tercera, que consistía en la posibilidad de modificar la trama del cableado que conectaba los contactos entre las dos caras de un mismo rotor.

Lo cierto es que el modelo original de Enigma experimentó grandes mejoras a lo largo de la contienda. Así, por ejemplo, mientras que el modelo Wehrmacht del Ejército y la Fuerza Aérea alemanes incluía cinco rotores y una hendidura, el utilizado por la Marina incluía ocho rotores y dos muescas o hendiduras. Más aún, al final del último rotor se incluyó un elemento llamado *reflector*, cuyo cometido consistía en conducir el proceso de cifrado por el camino inverso. Es decir, el resultado del último rotor se modificaba de nuevo a través de los rotores regresando desde el último situado a la izquierda hasta el primero de ellos situado a la derecha. El resultado era una máquina en la que el proceso de cifrado era igual al de descifrado, y con la que ninguna letra podía ser cifrada consigo misma.





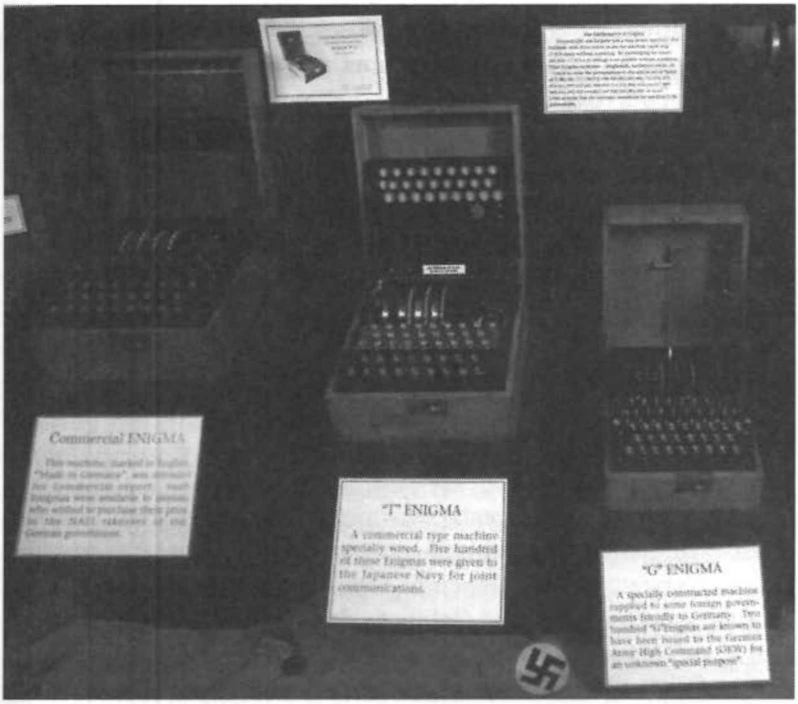


FOTO SUPERIOR IZQUIERDA:
Soldados alemanes transmitiendo mensajes con una máquina Enigma en el transcurso de la Segunda Guerra Mundial.

FOTO SUPERIOR DERECHA:
Alan Turing fotografiado en 1951.

FOTO INFERIOR:
Diferentes modelos de máquina Enigma.

Obviamente estas características fueron aprovechadas por los criptógrafos británicos en Bletchley Park, donde se construyó un auténtico complejo dedicado a descifrar los mensajes alemanes radiados interceptados. Además del reflector ubicado a la izquierda de los rotores, a la derecha de estos se situaba la rueda de entrada o *estátor*, que tenía como función conectar el teclado donde se escribía el mensaje de entrada con las lámparas que escribían el mensaje de salida ya cifrado. En el frontal de la máquina se incluyó un panel con un sistema de clavijas que convertían una letra en otra antes de que la letra original fuera transformada por los rotores. Si tenemos en cuenta todos los dispositivos que participaban en la transformación de una letra (el panel con el sistema de clavijas, los rotores y el reflector), el número de configuraciones posibles vendría dado por el producto de las permutaciones de los distintos dispositivos, alcanzándose la increíble cifra de 10^{114}. Un número realmente impresionante de configuraciones si consideramos que el cerebro humano tiene 10^{11} neuronas y el número de átomos del universo se estima próximo a 10^{80}. Con una máquina tan formidable, la Alemania del III Reich se sentía confiada, creyendo que la transmisión de los mensajes radiados, que contenían las órdenes entre las tropas y los mandos, estaba de sobra asegurada. Sin embargo, acontecimientos fortuitos jugaron en contra de la supuesta infalibilidad de Enigma y a favor de los Aliados, ya que entre otros acontecimientos, se capturaron algunas máquinas Enigma y sus libros de códigos en varios submarinos alemanes.

«BOMBAS» CONTRA ENIGMA

Uno de los países más castigados por la Alemania nazi, Polonia, se hizo de manera rocambolesca con una máquina Enigma enviada a Varsovia desde Alemania. Gracias a este hecho fortuito, un grupo de matemáticos del Gabinete de Criptología Biuro Szyfrów (BS 4) del Estado Mayor Polaco, bajo la dirección de Marian Rejewski (1905-1980), fueron capaces de descifrar los mensajes co-

dificados con Enigma, reenviando días después la máquina a su destinatario. Los polacos, y a través de ellos los Aliados, quedaron perplejos ante tan asombroso hallazgo: los alemanes codificaban la posición inicial de los rotores en el propio mensaje. La debilidad del método se debía a que la posición inicial del rotor, y por tanto, cuál de las veintiséis letras debía ser mostrada a través del visor, era indicada por duplicado en el mensaje. Por ejemplo, si la posición inicial del rotor era la letra B, entonces en el mensaje este dato se mostraba como BB. Desde 1932 Rejewski y su equipo descifraron con éxito los mensajes interceptados a los alemanes, pues fuera cual fuera la estrategia utilizada por estos últimos, siempre mantenían el criterio de repetición de letras.

Los matemáticos polacos construyeron una máquina, el ciclómetro, cuyo trabajo emulaba dos máquinas Enigma sincronizadas. Más tarde inventaron una nueva máquina criptoanalítica, a la que los polacos bautizaron con el nombre de *Bomba*, que era capaz de detectar ciertos patrones en un mensaje a través de una serie de rotores que emulaban el funcionamiento de tres máquinas Enigma. A partir del análisis de la frecuencia de estos patrones de letras en los mensajes, a los que llamaron *huellas* y *hembras*, lograron automatizar el descifrado de los mensajes interceptados.

Pero este éxito duró poco, ya que a finales de 1938 los alemanes introdujeron tres rotores más a Enigma, con lo cual pasaron a tener un total de seis rotores. Ahora los polacos necesitaban unas sesenta máquinas Bomba para poder descifrar con éxito un mensaje interceptado. La falta de recursos económicos les llevó a una inteligente decisión: en 1939 pasaron el testigo del contraespionaje a la inteligencia británica y francesa. Los británicos aceptaron el reto, creando la GC&CS (British Government Code & Cypher School), con sede en Bletchley Park, cerca de Milton Keynes, una ciudad próxima a Londres. No obstante, un grupo de criptógrafos polacos pertenecientes al BS 4 se dirigió a Francia. Una vez allí colaboraron hasta finales de 1942 con los servicios secretos franceses descifrando mensajes alemanes, que a su vez eran transmitidos a Bletchley Park. Después de que los alemanes ocuparan el sur de Francia, la mayor parte de los criptógrafos polacos se dirigieron al Reino Unido atravesando España. Lamentable-

mente, en opinión de algunos historiadores polacos, el talento de sus matemáticos no fue aprovechado por los británicos.

TURING EN BLETCHLEY PARK

Con cerca de diez mil trabajadores a finales de la Segunda Guerra Mundial, Bletchley Park fue un auténtico complejo de espionaje contra la Alemania nazi. Organizados en sectores ubicados en barracones, como si de una fábrica se tratase, los británicos se repartieron el trabajo de espionaje. En un sector los técnicos y analistas interceptaban los mensajes del Gobierno alemán o de sus ejércitos; otro sector se encargaba de descifrar los mensajes; mientras que un tercer sector, a partir de los mensajes descifrados, intentaba reconstruir el escenario o las intenciones de las operaciones militares de los alemanes. El trabajo se repartía teniendo en cuenta que los alemanes utilizaban distintas redes de comunicación, configurando las máquinas Enigma de forma distinta en cada una de estas redes. Con tal propósito el personal de Bletchley Park identificaba cada una de las redes con nombres en clave, como *Red* (rojo), *Shark* (tiburón) o *Chaffinch* (pinzón).

En el barracón número 8 (*Hut* 8) estaba Alan Turing, que se incorporó al complejo de Bletchley Park el 4 de septiembre de 1939, justo al día siguiente de que su país declarase la guerra a Alemania. Su misión era descifrar los códigos Enigma de la Marina alemana con el fin de romper el bloqueo naval que llevaban a cabo los submarinos *U-Boot*. Según palabras del historiador británico Asa Briggs (n. 1921), quien también sirvió en Bletchley Park desde 1942 hasta 1945 en el barracón 6, contaron con la participación de personas de gran talento, y de entre todas, el genio sin excepción fue Alan Turing. Durante esta época Turing viajó a Estados Unidos para hacer de puente entre ambos países aliados. Al parecer parte de su trabajo consistió en diseñar un sistema de cifrado para las conversaciones telefónicas entre los máximos mandatarios de cada país, Roosevelt y Churchill. Para ese cometido Turing contó con la colaboración de Dilly Knox (1884-1943), un

criptógrafo educado en el King's College de Cambridge, con el que trabajó también conjuntamente en la tarea de descifrar los mensajes de Enigma de forma rápida y automática. El método propuesto resultó ser más eficaz que el de los matemáticos polacos, cuyos conocimientos sobre Enigma recogieron en la obra *Treatise on Enigma (Tratado sobre Enigma)*.

En esta época Turing, apodado por sus colegas como *The Prof*, abreviatura de profesor en lengua inglesa (en español el vocablo inglés *professor* significa catedrático de universidad), llamaba la atención por algunas de sus excentricidades. Por ejemplo, acostumbraba a atar su taza a los tubos de la calefacción para evitar que se la robasen, o como se recoge en alguna de sus biografías, en ciertas ocasiones acostumbraba a ir corriendo desde Bletchley hasta Londres, unos 64 kilómetros, para participar en reuniones de trabajo.

Ante el curso de los acontecimientos bélicos, los británicos optaron por el diseño de una nueva máquina heredera de la polaca Bomba, a la que bautizaron con el nombre de *Bombe*. Se trataba de un sistema electromecánico que emulaba el trabajo conjunto de un grupo de máquinas Enigma. La versión original fue ideada por Alan Turing en 1939 en Bletchley Park y construida por Harold Keen (1894-1973), de la BTM (British Tabulating Machine Company), una empresa emparentada con la que en Estados Unidos sería tiempo después IBM. Por aquel entonces estas empresas se dedicaban, a ambos lados del Atlántico, a comercializar máquinas tabuladoras o censadoras. Con la máquina inventada por el estadístico estadounidense Herman Hollerith (1860-1929) podían leerse tarjetas perforadas, usadas para realizar los censos de población, ya que perforando una u otra posición en una cartulina era posible codificar las respuestas. Al parecer, según Edwin Black, autor de *IBM y el holocausto* (2001), Adolf Hitler adquirió a IBM las máquinas tabuladoras con las que en 1933 censó a la población judía en Alemania, y por ello su fundador, Thomas J. Watson (1874-1956), recibió en 1937 la Cruz al Mérito del Águila Germana, que le fue entregada por el propio Führer en Berlín.

Con posterioridad el modelo original de Bombe fue mejorado por Gordon Welchman (1906-1985), y por eso el modelo final es

conocido como *Bombe de Turing-Welchman*, distinguiéndose así de paso de su antecesor, la máquina polaca denominada *Bomba kryptologiczna*. Una máquina Bombe pesaba cerca de una tonelada e incluía 108 rotores agrupados de tres en tres, emulando los tres rotores de Enigma. A su vez, los grupos de tres rotores se agrupaban por docenas, es decir, la máquina estaba formada por tres secciones de 12 grupos de tres rotores. Todos estos rotores realizaban un trabajo similar a los de Enigma pero en sentido inverso, descifrando mensajes. Desde un punto de vista mecánico,

EL PROYECTO SIGSALY

Desde finales de 1942 hasta la primavera de 1943 Turing estuvo en Estados Unidos. En una visita a los Laboratorios Bell conoció al célebre Claude Shannon (1916-2001), padre de la teoría de la información. Aunque Turing estaba entusiasmado por poder hablar con tal celebridad acerca de la posibilidad de construir un «cerebro artificial», su cometido en dicha visita era otro: recoger ideas para trabajar sobre un sistema de encriptación de la voz que protegiese las conversaciones telefónicas entre los máximos mandatarios de ambos países, Roosevelt y Churchill. El proyecto se denominó SIGSALY. El sistema encriptaba la voz por medio de lo que se llama ruido aleatorio y fue muy utilizado por los Aliados durante la contienda. Como curiosidad, cabe mencionar que SIGSALY aparece en la novela de ciencia ficción *Cryptonomicon* (1999), de Neal Stephenson (n. 1959), en una conversación ficticia entre uno de los personajes de la novela, Lawrence Waterhouse, y Alan Turing. Una vez finalizada la guerra, Turing abandonó Bletchley Park para trabajar en el HMGCC (Her Majesty's Government Communications Centre), donde participó en el diseño y construcción de un sistema portátil para cifrar la voz llamado *Delilah*. Para realizar una demostración del correcto funcionamiento del sistema, tanto del proceso de cifrado como del de descifrado, utilizó una grabación de la voz de Winston Churchill.

El proceso del cifrado de la voz

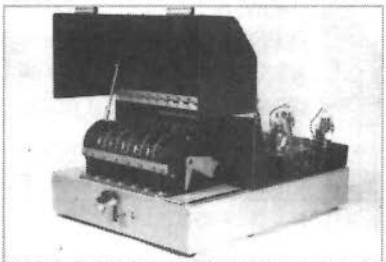
El cifrado de la voz requiere de un proceso que consta de varias etapas. En primer lugar se procede a muestrear el sonido. Para muestrear la voz grabamos pequeños fragmentos de sonido en diferentes tiempos, es decir, con una cierta tasa de muestreo. En los sistemas de telefonía actual, las frecuencias por debajo de 4000 Hz son las que se transmiten en una conversación, dato que es importante a la hora de ajustar la tasa de muestreo en una conversación

los rotores tenían el mismo cableado interno que Enigma; el reflector era simulado con una idea muy simple: los contactos y cables estaban por duplicado. Una vez que era interceptado un mensaje cifrado radiado, pasaba a ser el *input*, y Turing decidía cuál debía ser el cableado entre los grupos de tres rotores por el que circularía el mensaje hasta ser descifrado, lo que en lenguaje actual llamaríamos *output*.

En Estados Unidos también se construyeron para el Ejército máquinas que realizaban tareas similares, aunque con un diseño

telefónica. En segundo lugar, el sonido capturado, la voz, es tratada mediante un proceso llamado *normalización*. Este proceso es necesario como un paso previo para asegurarse de que las diferencias en los sonidos que hemos capturado son de intensidad y no «ruido» o cualquier otro artefacto causado, por ejemplo, por el propio sistema de telefonía. Además, las personas, por hábitos de articulación, pronunciamos de manera distinta las vocales y con distinto volumen; con el proceso de normalización eliminamos estas diferencias. A continuación, la voz ya normalizada es por fin encriptada. En el procedimiento ideado por Turing los fragmentos de voz eran normalizados a una escala entre 0 y 1. Una vez normalizados, los fragmentos eran transformados con el operador aritmético «módulo» (mod). Este operador da como resultado el resto de una división entera; por ejemplo, 5 mod 2 será 1. Finalmente, la onda de voz así transformada era reconstruida siguiendo un proceso inverso. Pese a este éxito el sistema nunca fue utilizado. Lo cierto es que la participación de Turing en ambos proyectos quedó en un segundo plano, pese a su importancia histórica y política, ante sus éxitos en muchos otros temas de investigación.



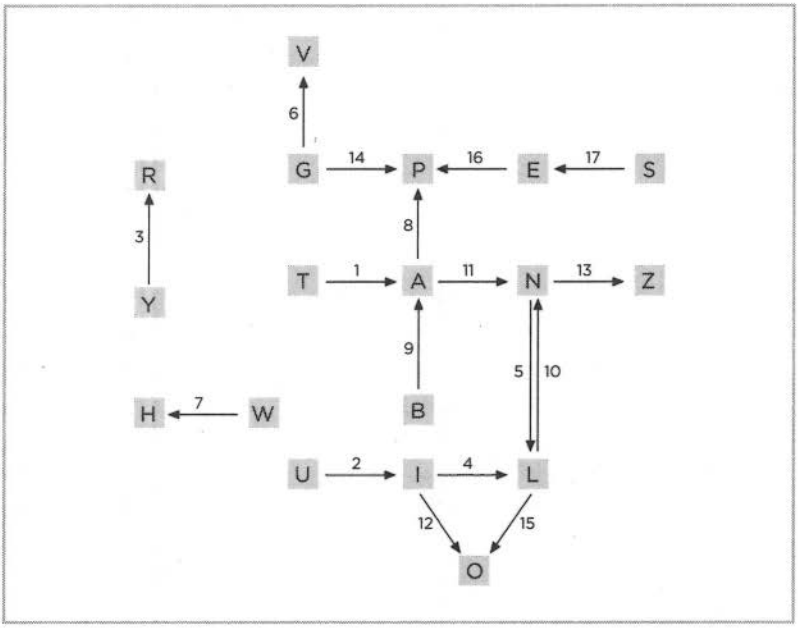


Máquinas para el sistema de encriptado de voz Delilah.

distinto. Según los estadounidenses, sus máquinas eran más rápidas y las secuencias crib más cortas que las utilizadas por los ingleses. La versión estándar construida por los británicos era equivalente a 36 máquinas Enigma e incluso podía llegar a descifrar dos o tres mensajes simultáneamente. Cada vez que se iba a descifrar un mensaje, la máquina requería un menú, que utilizaba lo que los ingleses llamaban *crib*. Con este nombre hacían referencia a un ejemplo de texto o mensaje sin cifrar, del que además se disponía de su versión cifrada, por ejemplo, algún fragmento de texto cifrado y sin cifrar que había sido capturado al enemigo. Para que el crib fuera efectivo, era necesario conocer muy bien la jerga militar alemana y el protocolo que estos seguían en el envío de mensajes. Fue de mucha ayuda saber que Enigma nunca cifraba una letra, por ejemplo A, utilizando una letra similar, A en el ejemplo. Una vez elegido el crib, el operador de Bombe diseñaba un menú tal y como se ilustra en la tabla siguiente. Supóngase que el crib es TURINGHABLAINGLES y el texto cifrado (fila identificada en la tabla como TC) AIYLLVWPANNOZPOPE. Con el fin de que el ejemplo represente una situación lo más real posible hemos utilizado un simulador de la máquina Enigma para obtener el texto cifrado (http://www.bletchleypark.org.uk/content/simulator.html). A partir de los dos mensajes construiremos una tabla en la que se asocie para cada letra del texto cifrado la que le corresponde en el mensaje original o crib:

CRIB	T	U	R	I	N	G	H	A	B	L	A	I	N	G	L	E	S
TC	A	I	Y	L	L	V	W	P	A	N	N	O	Z	P	O	P	E
P	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
TS	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
TM	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
TI	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

También anotaremos en la tabla la posición de cada letra en el texto del mensaje (lo que designaremos como P). A continuación se representa un diagrama en el que se muestran las asociaciones entre letras y en qué posición del texto están:



A partir de este grafo el operador era capaz de diseñar el menú con el que configurar Bombe, así como las posiciones iniciales de los tambores localizados en la parte superior (TS), media (TM) e inferior (TI). Una vez realizada la configuración, la máquina ejecutaba su trabajo deteniéndose cada vez que encontraba una solución candidata, es decir, un mensaje descifrado. En la figura anterior se observan algunos bucles, como por ejemplo, ILO. Un detalle curioso es que Turing observó que cuanto mayor era el número de bucles, menor era el número de paradas, y por tanto menor el número de mensajes descifrados erróneamente. El aspecto general de Bombe era bastante atractivo, ya que los tambores fueron pintados en diferentes colores, que representaban el rotor de Enigma que emulaban. Cada tambor podía estar en una de las veintiséis posiciones posibles y por tanto el total de configuraciones de los tres tambores era $26 \times 26 \times 26 = 17576$. Cada vez que Bombe encontraba una posible solución se detenía. Lo normal era que se detuviera varias veces, dando como resultado mensajes descifrados erró-

neos, hasta que lograba dar con el mensaje descifrado correcto. Un paso fundamental del criptoanálisis consistía en asegurarse de que el mensaje descifrado era realmente la solución correcta. Para ello cifraban de nuevo el mensaje descifrado con TypeX, una máquina británica que emulaba a Enigma, estudiando el resultado obtenido.

El primer modelo de Bombe fue construido el 18 de marzo de 1940. A finales de la Segunda Guerra Mundial los británicos disponían de 211 máquinas Bombe en Bletchley Park, para cuyo mantenimiento y uso contaban con unas dos mil personas. Gracias al rotundo éxito logrado con estas máquinas, nació la leyenda de Alan Turing. Su trabajo como criptógrafo, y el de todo el complejo de Bletchley Park, tuvo una gran influencia en el desarrollo de la contienda. En la actualidad se sabe que gracias a su contribución se conocieron las fechas de los ataques aéreos contra Inglaterra y las rutas de los submarinos y navíos alemanes; también contribuyó a la victoria en África contra el mariscal Rommel y facilitó las operaciones militares de los Aliados en el oeste de Europa.

No cabe duda de que el diseño de Bombe fue una de las grandes contribuciones de Turing como criptógrafo durante la guerra, pero no fue la única. También desarrolló procedimientos estadísticos para un uso más eficaz de Bombe, que fueron muy útiles a la hora de descifrar los mensajes de la Marina alemana codificados con Enigma. Estas técnicas recibieron el nombre de *Banburismus*. Además, introdujo otro procedimiento, denominado *Turingery* o *método de Turing*, mediante el que era posible descifrar mensajes cifrados con otra máquina infernal, la máquina de Lorenz SZ 40/42. Asimismo, hacia finales de la contienda, Turing desarrolló, en esta ocasión para el HMGCC (Her Majesty's Government Communications Centre), un sistema portátil bautizado en clave con el nombre de *Delilah*, con el que era posible cifrar las conversaciones telefónicas.

Al terminar la contienda, el primer ministro británico, Winston Churchill, mandó destruir todas las máquinas Bombe y los documentos relacionados. En esa etapa de su vida Turing hizo de enlace entre Estados Unidos y el Reino Unido. Fue precisamente allí cuando comenzó a pensar en la posibilidad de construir una «máquina inteligente», lo que más tarde conduciría a su trabajo pionero en inteligencia artificial. También fue en esa época en la que



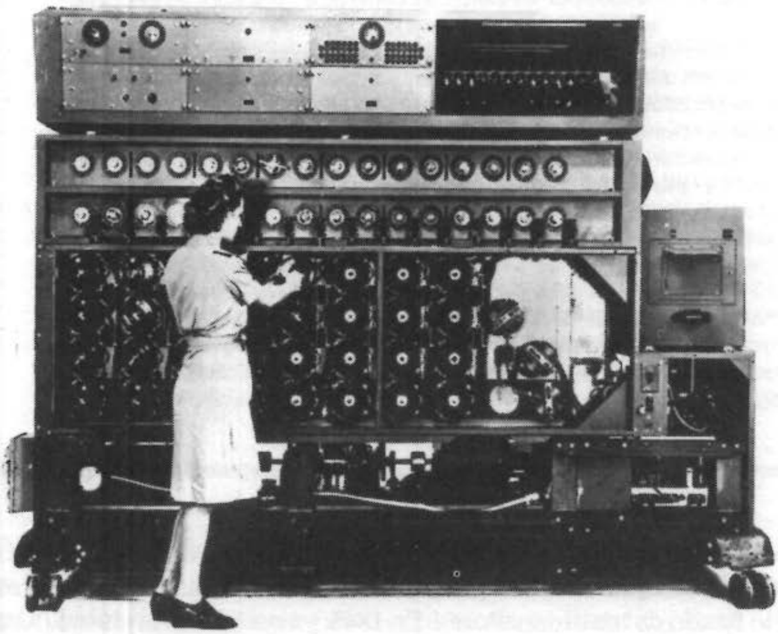
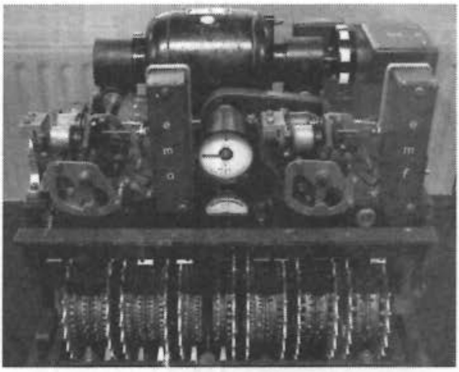


FOTO SUPERIOR:
Uno de los barracones de Betchley Park, donde personal especializado trabajaba para descifrar el código Enigma.

FOTO INFERIOR:
La máquina Bombe, ideada por Alan Turing y construida en Betchley Park por Harold Keen.

LORENZ, LA OTRA MÁQUINA INFERNAL



Máquina de Lorenz SZ42.

A principios de la Segunda Guerra Mundial los británicos interceptaron unas señales provenientes del bando alemán que, para su sorpresa, no utilizaban el código Morse ni estaban codificadas con Enigma. Se trataba de señales codificadas con Lorenz, otra máquina para cifrar mensajes que, conectada a un teletipo, resultaba tan útil como Enigma. En Bletchley Park todas aquellas señales alemanas que circulaban a través de teletipo recibían el nombre en clave *Fish* (pez), y las que estaban codificadas con Lorenz SZ 40/42 recibían el código específico *Tunny* (atún). Una vez más, como ya había sucedido con Enigma, una mezcla de suerte y errores cometidos por los alemanes permitió a los británicos averiguar cómo funcionaba Lorenz, esta vez sin haber tenido jamás una máquina en sus manos. Esto permitió que en Bletchley Park se construyera una máquina electromecánica con una lógica similar, mediante la que era posible descifrar los mensajes codificados con Lorenz, denominada *Tunny machine* (máquina atún). Pero ¿cómo funcionaba Lorenz SZ 40/42? En primer lugar, cada vez que se escribía un carácter, este era transformado a otro del código Baudot, inventado en 1874 y que desde entonces se utiliza en telegrafía: consistía en que un carácter era representado por una secuencia de cinco unos y ceros, es decir, 5 bits. En aquella época el 1 y el 0 eran representados en la cinta, respectivamente, como un «agujero» o un «espacio sin agujerear». La máquina de Lorenz incluía un ingenioso sistema mecánico en el que un conjunto de ruedas hacían la tarea de lo que hoy se denomina *generador de números aleatorios*, una clase de algoritmo con el que es posible obtener números aleatorios en un ordenador y que es muy utilizado en toda clase de sorteos, videojuegos, simulaciones, criptografía, etc. El método utilizado por la máqui-

se familiarizó con la electrónica, y tal vez fue allí en Bletchley Park donde comprendió la importancia de esta disciplina en el desarrollo futuro de los ordenadores. En 1945, ya concluida la guerra, Alan

na de Lorenz para cifrar un mensaje consistía en generar una secuencia aleatoria de 5 bits con una serie de doce ruedas dentadas (*pinwheels* en inglés), cada una de las cuales poseía en su perímetro un número determinado de pernos. Estos pernos podían colocarse en dos posiciones: *on*, o 1, o *off*, o 0, de forma que al girar se generaba una secuencia de unos y ceros, o sea, bits. Si un perno estaba en *on* entonces se invertía el valor del bit correspondiente de la letra a codificar, de 0 a 1 y de 1 a 0, y cuando el perno estaba en *off*, se conservaba su valor. A continuación se aplicaba el operador booleano XOR («exclusivo OR») entre cada uno de los bits del carácter y del carácter modificado. La tabla de este operador es la siguiente:

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Esta receta se aplicaba de modo secuencial, varias veces, hasta conseguir transformar el carácter inicial en otro del código Baudot. Por ejemplo, si deseáramos cifrar el apellido TURING, el primer paso sería representarlo en código Baudot; así obtendríamos la secuencia 10000-00111-01010-00110-01100-11010. Supóngase que hubiéramos cifrado la secuencia de caracteres TURIN y procediéramos finalmente a cifrar la última letra del apellido. En segundo lugar, si el operador de la máquina hubiera configurado los pernos de una rueda, a la que llamaremos R1, como on-on-off-off-on entonces la secuencia 11010, que representa a la letra G, invertirá el valor de los bits que estuvieran afectados por los pernos en estado on de la rueda. El resultado es que la secuencia 11010 que representa a G se transforma en 00011, secuencia que corresponde a la letra A. Seguidamente, repetiremos estos pasos una vez más. Supongamos que el operador de la máquina ha configurado una segunda rueda, a la que denominaremos R2, disponiendo los pernos como on-off-on-off-on. En tal caso esta última rueda transformará la secuencia 00011, convirtiéndola en la secuencia 10110, que en código Baudot corresponde al carácter P. Por tanto, con la máquina de Lorenz habremos cifrado la letra G como P.

Turing fue galardonado con la Orden del Imperio Británico. Por fin se reconocía su genialidad, que puso al servicio de la victoria de los Aliados con su trabajo como criptógrafo en Bletchley Park.

COLOSSUS: EL NACIMIENTO DEL ORDENADOR

Los avances científicos y tecnológicos son en muchos casos el resultado «positivo» de los conflictos bélicos, y así fue como ocurrió con Colossus. Con este nombre se bautizó en Bletchley Park a la primera máquina electrónica programable, que, pese a algunas limitaciones, podría denominarse ordenador. Si Bombe fue la respuesta a Enigma, Colossus lo fue a Lorenz SZ 40/42. La máquina de Lorenz cifraba los mensajes utilizando una secuencia de números aleatorios. Tales números eran obtenidos aplicando un método electromecánico basado en una serie de ruedas dentadas *(pinwheels)*. Afortunadamente, los números obtenidos carecían de la aleatoriedad que, por ejemplo, acompaña a un número extraído de un bombo de lotería, y se producían ciertos patrones en las secuencias de números. Este hecho fue de gran ayuda para descifrar con éxito los mensajes interceptados por los Aliados.

En realidad el «corazón» de Colossus no era nuevo, sino que lo heredó de las máquinas Robinson. Con este nombre se bautizó a una familia de máquinas diseñadas para descifrar los mensajes cifrados con Lorenz. Una máquina Robinson utilizaba dos cintas, una con el mensaje cifrado y la otra con una secuencia de números aleatorios que habían sido obtenidos con un sistema de ruedas similar al de una máquina de Lorenz. La mejora introducida en Colossus fue la sustitución de la segunda cinta —la secuencia de números aleatorios— por circuitos electrónicos de válvulas. Un gran inconveniente de las máquinas Robinson era que con cierta frecuencia solía romperse de forma accidental la segunda cinta, debido a que se requería una alta velocidad de lectura de los números aleatorios. Este contratiempo se evitaba con Colossus, que era capaz de leer unos 5 000 caracteres por segundo, todo un hito para la época. Aunque Alan Turing no participó en su diseño, Colossus fue ideada por uno de los mentores de Turing, Max Newman, y otros colegas de Bletchley Park.

La primera versión de este «ordenador» fue obra de Tommy Flowers (1905-1998), un técnico de la British Post Office Research Station. Una de las ideas innovadoras que permitió construirlo fue la propuesta de Flowers de utilizar válvulas electrónicas, las mis-

SECUENCIAS DE NÚMEROS ALEATORIOS

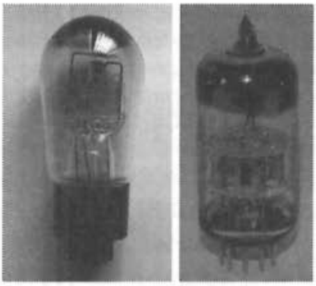
Un ordenador es una máquina de Turing universal y esto significa que a partir de un cierto estado del ordenador y de unos ciertos datos de entrada realizará una serie de tareas u operaciones que conducirán a un resultado completamente previsible. Por ejemplo, si en una hoja de cálculo diseñada para calcular un presupuesto introducimos unos determinados datos numéricos, o *input*, el resultado, u *output*, será siempre el mismo. Uno de los retos científicos más interesantes desde la época de John von Neumann, uno de los primeros investigadores que planteó esta cuestión en simulaciones para el diseño de las primeras bombas atómicas, fue el diseño de algoritmos capaces de generar una secuencia de números que se asemejase a la secuencia que obtendríamos si tales números hubieran sido obtenidos con un bombo de lotería. Puesto que los números procedentes de un bombo son números aleatorios, los obtenidos con ordenador recibieron el nombre de *pseudoaleatorios*; un programa de ordenador capaz de producir esta clase de números se denomina *generador de números aleatorios*. Los números pseudoaleatorios están siempre comprendidos en el intervalo [0, 1]. Por ejemplo, la siguiente secuencia de doce números 0.092833, 0.472751, 0.542341, 0.022788, 0.069853, 0.317325, 0.808213, 0.225401, 0.633599, 0.133044, 0.530186, 0.477541 ha sido obtenida con el siguiente programa en BASIC-256:

```
n=0
do
   u=rand
   print u
   n=n+1
until n=12
```

Pongamos otro ejemplo: ¿cómo simularíamos un dado con el programa anterior? Sencillamente sustituiremos `u=rand` por `u=int(rand*6)+1`. Los números obtenidos con un programa deben cumplir algunas propiedades, en particular, estar comprendidos entre 0 y 1, ser independientes unos de otros, esto es, si obtenemos un cierto número, por ejemplo 0.808213, dicho valor no debería influir en el siguiente número que se genere de la secuencia, 0.225401, y además debe cumplirse que todos los números tengan la misma probabilidad de ser obtenidos. Un aspecto curioso es que los números a título individual no son aleatorios, pero sí la secuencia de la que forman parte, cuyas propiedades estadísticas son similares a las que tiene una secuencia de números obtenidos con un sistema mecánico de lotería. No obstante, ya existe la posibilidad de obtener a través de Internet auténticos números aleatorios en el ordenador a partir de fenómenos físicos, en lugar de un algoritmo como el que se utiliza en la función `rand` de BASIC-256.

mas utilizadas en los circuitos de las primeras radios, naciendo así el primer ordenador electrónico de la historia. Colossus llegó a tener nada más y nada menos que unas 1500 válvulas. Cabe decir que estos componentes electrónicos fueron utilizados en los ordenadores construidos antes de 1959. Tanto la primera versión de Colossus, llamada *Mark 1*, como otra mejorada, o *Mark 2*, entraron en funcionamiento en 1944. Podríamos afirmar que, mientras que Alan Turing trabajó sobre la lógica que subyace en los orde-

VÁLVULAS ELECTRÓNICAS Y PUERTAS LÓGICAS



Dos clases de válvulas: diodo (izquierda) y tríodo (derecha).

Una válvula electrónica es un tubo en el que se ha hecho el vacío, que contiene un filamento emisor de electrones, el cátodo (carga negativa), y una lámina metálica receptora de los electrones, el ánodo (carga positiva). El resultado es una corriente de electrones desde el cátodo —una vez está incandescente— hasta el ánodo. Puesto que la corriente circula en un solo sentido, la válvula descrita realiza la función de uno de los componentes electrónicos más importantes, el diodo. Posteriormente, en el diodo se intercaló un filamento adicional entre el cátodo emisor de electrones y el ánodo receptor. Cuando se aplicaba una corriente al filamento adicional, se ejercía un control sobre el flujo de electrones que circulaba desde el cátodo hasta el ánodo, amplificándose el voltaje. Con este añadido se inventó el triodo, componente electrónico que realizaba la misma función de lo que hoy se denomina *transistor*. Con estos componentes electrónicos es posible construir circuitos que realicen operaciones aritméticas, por ejemplo sumar dos números, y lógicas, por ejemplo comparar dos cifras determinando cuál es la mayor.

Ceros y unos

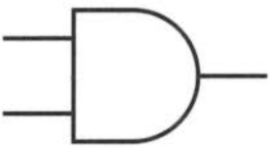
En los ordenadores, incluido Colossus, las operaciones aritmético-lógicas se realizan por medio de lo que se conoce como álgebra de Boole, que opera con bits, y por tanto con los dígitos 0 y 1, aplicándoles operadores denominados *puertas* en lenguaje de electrónica. Pongamos un ejemplo. Supongamos que una corriente de 0 V representa al dígito 0 y una de 3 V al dígito 1. Por

nadores, Tommy Flowers diseñó el *hardware* y, por tanto, los circuitos electrónicos que dan vida a la lógica de un ordenador.

Uno de los circuitos más ingeniosos de Colossus utilizaba dos clases de válvulas, los tiratrones y los fotomultiplicadores, con las que era capaz de leer los caracteres de una cinta de papel. Con el tiratrón era posible grabar 1 bit; conectando varias de estas válvulas entre sí, los ingenieros de Bletchley Park construyeron la memoria del ordenador. El fotomultiplicador era una válvula cuyo

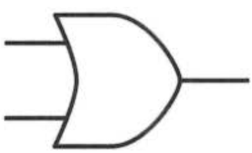
consiguiente, el hecho de que «circule» o «no circule» corriente eléctrica define el valor 0 o 1 de un bit, que es la cantidad más pequeña de información que puede procesar un ordenador. Una puerta es un circuito electrónico con diodos o transistores en el que una entrada, 0 o 1, se transforma en una salida, también 0 o 1, como resultado de la aplicación de algún operador del álgebra de Boole. De todos los operadores posibles, AND y OR son dos de los más utilizados en electrónica digital. La puerta AND, equivalente a nivel lógico a la conjunción «y», es aquella en la que la salida es 1 si todas las entradas reciben 1 a la vez. Por el contrario, la salida será 0 si una o las dos entradas reciben 0. La tabla y el símbolo que resumen esta puerta son:

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1



Una puerta OR equivale a la conjunción disyuntiva «o»; en este caso, la salida será 1 tanto si una, la otra o ambas de las entradas reciben un 1. La tabla y el símbolo que resumen esta puerta son:

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1



funcionamiento era similar al de la célula fotoeléctrica: producía una señal amplificada en el ánodo de la válvula cada vez que recibía un haz de luz. Con estos componentes electrónicos, la capacidad de Colussus durante la Segunda Guerra Mundial era equivalente a un ordenador con microprocesador Pentium fabricado en 2004. Más aún, lo sorprendente es que Colossus solo incluía dos puertas booleanas, AND y OR, en sus circuitos.

Pese a que fue un logro de la ingeniería de su época, la programación de Colossus era muy primitiva comparada con la de los ordenadores actuales, ya que para escribir un programa era necesario configurar numerosas clavijas e interruptores. Ahora bien, pese a tratarse de una máquina programable, Colossus no fue un ordenador en un sentido estricto, ya que no era una máquina de Turing universal. Es decir, no podía ser programado para realizar otras tareas que no fueran «romper códigos» escritos con una máquina de Lorenz SZ 40/42 y, por tanto, no era una máquina de propósito general. Además, actualmente un ordenador es una máquina de Turing universal que puede ser programado en distintos lenguajes de programación (C, Java o Visual Basic, por ejemplo). Concluiremos por tanto que Colossus fue casi un ordenador, parcialmente programable, puesto que solo era útil para lo que fue diseñado, y en consecuencia no universal. La coincidencia de Alan Turing y Colossus en un mismo tiempo y lugar representó para el científico inglés una experiencia tan estimulante que le llevó a estudiar electrónica y a considerar la posibilidad de construir un auténtico ordenador. Fue allí, en Bletchley Park, donde nació el sueño de construir una máquina de Turing universal, empresa que logró con éxito tiempo después con el diseño y construcción del ordenador Pilot ACE.

Finalizada la contienda mundial, por motivos de seguridad militar Winston Churchill mandó destruir todas las máquinas Colossus y quemar los planos en los que se describían su diseño y sus circuitos. Tan ingrata tarea fue realizada por su creador, Tommy Flowers, que indultó dos máquinas posteriormente utilizadas durante la Guerra Fría, y finalmente destruidas en la década de los años sesenta. El éxito de Bletchley Park con Colossus no salió a la luz pública hasta 1976, cuando la Ley sobre Secretos Oficiales

permitió hacerlo público. Durante años ENIAC, construido en Estados Unidos en 1946, era considerado el primer ordenador electrónico de la historia. Actualmente, tras una revisión de la historia, Colossus, construido en 1944, ocupa ese lugar de honor. En el Museo Nacional de Computación, ubicado en Bletchley Park, se exponen al público dos réplicas de Colossus construidas en 1996 y 2004 bajo la dirección de Tony Sale (1931-2011), ingeniero en electrónica e historiador de la informática. Gracias a los mensajes descifrados con Colossus se supo que Hitler estaba desengañado ante los acontecimientos bélicos y que creyó que el desembarco de los Aliados tendría lugar por el Paso de Calais, por lo que ordenó que sus divisiones Panzer se dirigieran hacia allí. La pesadilla terminó finalmente en la primavera de 1945 con su suicidio en el búnker de Berlín. No obstante, el verano de ese mismo año vio la luz una nueva pesadilla para la humanidad: el lanzamiento de dos bombas atómicas en las ciudades japonesas de Hiroshima y Nagasaki. Una nueva era, la Guerra Fría, comenzaba tras la Segunda Guerra Mundial: el mundo quedaba repartido así en dos grandes bloques, el occidental-capitalista y el oriental-comunista, que estarían enfrentados desde 1945 hasta 1989, con la caída del Muro de Berlín.

CAPÍTULO 3

Los primeros ordenadores ¿británicos o estadounidenses?

Durante su estancia en Bletchley Park Turing fue testigo del nacimiento de Colossus. Este hecho representó un estímulo para que diseñara su primer ordenador, el Pilot ACE, según sus ideas y especificaciones. A mediados de los años cuarenta y principios de los cincuenta la construcción de varios modelos de ordenadores a uno y otro lado del Atlántico condujo a una polémica aún abierta sobre qué país fue realmente el pionero en el diseño y construcción de ordenadores.

Una vez concluida la Segunda Guerra Mundial, Alan Turing abandonó Bletchley Park y, como el resto de sus compañeros, tuvo que volver a la vida civil. Afortunadamente, recibió una invitación para incorporarse al Laboratorio Nacional de Física (NPL, por sus siglas en inglés), en Londres, un instituto dedicado al desarrollo de estándares en ciencia y tecnología, dirigido en ese momento por Charles Galton Darwin (1887-1962), nieto de Charles Darwin (1808-1882). La propuesta era que Turing se encargara de dirigir un proyecto pionero: el diseño y construcción de un ordenador. En 1946 Turing había enviado al NPL un informe con algunas ideas generales sobre cómo debía ser en la práctica un ordenador, máquina que John Womersley (1907-1958), colega de Turing responsable de la sección de matemáticas del instituto y quien realmente lo invitó a incorporarse, bautizó como *Automatic Computing Engine* (ACE). La palabra *engine* (motor en lengua inglesa) fue adoptada en homenaje a Charles Babbage (1791-1871), creador de la máquina analítica y la máquina diferencial, consideradas como precursoras de los actuales ordenadores. En ese informe Turing se adelantó a su tiempo: dio detalles tanto del *hardware*, es decir, de los circuitos electrónicos, como del *software*, esto es, los programas, especificando unas reglas generales sobre cómo escribir un programa para ser ejecutado en el ordenador ACE. Por fin iba a tener la oportunidad de dar el gran salto y

pasar de la teoría a la ingeniería, llevar a la práctica su trabajo sobre las máquinas de Turing universales. Su sueño se hacía realidad, y así nacería el ordenador Pilot ACE.

UN SUEÑO HECHO REALIDAD: PILOT ACE

En los ordenadores construidos a principios de los años cincuenta dos fueron los dispositivos que se utilizaron como memoria: el tubo de rayos catódicos y las columnas de mercurio, con las que se diseñaron las primeras memorias de línea de retardo. Los ingenieros de la época diseñaron una clase de memoria en la que, para almacenar los datos, se utilizaba como principio el tiempo que necesitaba una señal para propagarse por un medio físico, por ejemplo el mercurio. Este segundo dispositivo fue el que Turing eligió para Pilot ACE por su mayor velocidad de recuperación de datos y fiabilidad. La columna de mercurio disponía en cada extremo de un dispositivo piezoeléctrico, ideado a partir de un micrófono y altavoz, que hacía la función de transductor, convirtiendo en una onda ultrasónica (frecuencia sonora de aproximadamente 20 000 Hz) los pulsos eléctricos generados por el amplificador de un radar. Una vez que la onda llegaba a través del mercurio al otro extremo, era convertida de nuevo en impulsos eléctricos, que eran enviados a una pantalla. Otros ordenadores, como EDSAC, CSIRAC o UNIVAC I, también utilizaron columnas de mercurio. Así, por ejemplo, UNIVAC I, uno de los primeros ordenadores comerciales de los años cincuenta, disponía de siete «tanques de memoria», cada uno de ellos con dieciocho columnas de mercurio. La velocidad de acceso a los datos era de 222 microsegundos, un verdadero prodigio para la época, teniendo cada columna una capacidad de almacenamiento de diez palabras, por ejemplo las órdenes de un programa, con una longitud de doce caracteres.

Con posterioridad las líneas de mercurio fueron sustituidas por memorias de tambor, un dispositivo más avanzado. Se trataba de un cilindro metálico con una superficie que contaba con propie-

dades ferromagnéticas sobre la que se disponía una serie de cabezas de lectura y escritura. Esta clase de memoria se utilizó durante toda la década de 1950 y entre sus novedades destacaba el método utilizado para gestionar su uso, conocido como *entrelazado*: disponía los datos de manera no contigua, técnica aún utilizada por algunos discos duros, además de las transmisiones vía satélite o el ADSL. De hecho, con ese dispositivo de memoria, Pilot ACE llegó a almacenar hasta 4096 secuencias de unos y ceros. Desde entonces, en homenaje a este dispositivo de memoria, algunas versiones del sistema operativo Unix han dedicado el directorio `/dev/drum` (*drum*, tambor en inglés) al lugar en el que se gestiona la memoria virtual. Gracias a estas características, el ordenador ideado por Turing fue uno de los más avanzados de su época; su memoria llegó a tener una capacidad de almacenamiento muy similar a la de los primeros ordenadores Macintosh de Apple.

> «Turing sentía una gran predilección por trabajar sobre cualquier tema desmarcándose de los principios establecidos. Normalmente empezaba sin consultar los trabajos previos sobre la materia, y sin duda ese hábito fue el que le proporcionó a su trabajo ese aire tan característico de originalidad.»
>
> — Maurice V. Wilkes sobre Alan Turing, en *Ordenadores, antes y ahora*.

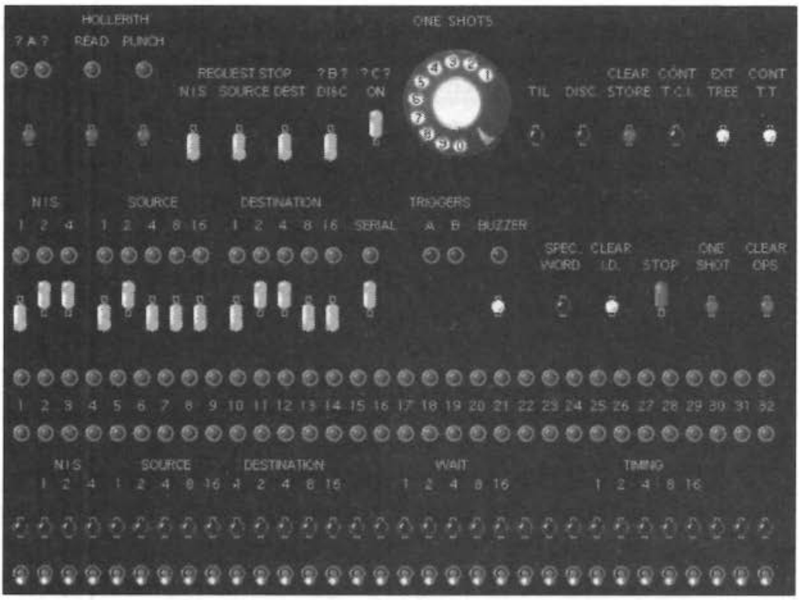
La gestión de la memoria fue otra de las contribuciones de Alan Turing al campo de la informática. Los datos se grababan por el llamado *método de dos direcciones*. La memoria de un ordenador se organiza desde un punto de vista «lógico», como si estuviera compuesta por celdas. La posición de cada celda está identificada por un número denominado *dirección de memoria*. Las órdenes con las que se escribe un programa —por ejemplo en lenguaje BASIC-256, `print`, `dim` o `input`— forman un texto, el código fuente —lo que «escribe» un programador—, que es almacenado en la memoria del ordenador una vez traducido a código máquina o binario, esto es, a una secuencia de unos y ceros.

En el ordenador diseñado por Turing cada orden tenía asociada tanto la posición, o dirección de memoria en la que se

encontraba almacenada, como la dirección de la siguiente instrucción a almacenar. Si en algo hizo hincapié el científico inglés fue en que su ordenador debía cumplir dos requisitos: ser suficientemente rápido ejecutando cualquier programa y disponer de una cantidad de memoria que resultara satisfactoria. Desde entonces todos los ordenadores que se han construido han intentado satisfacer estos dos requisitos. El *hardware* estaba construido con válvulas electrónicas, aproximadamente unas 800, un número no demasiado alto, motivo por el cual el ordenador resultaba bastante fiable, pues así se reducía el riesgo de que se fundieran una o más válvulas durante la ejecución de un programa y tuviese que ejecutarse de nuevo. Con una velocidad de 1 megahercio, fue uno de los ordenadores más rápidos del Reino Unido. Además, realizaba las operaciones aritméticas en coma flotante, es decir, era capaz, como los ordenadores actuales, de representar un número con muchos decimales en el siguiente formato: por ejemplo, $6{,}127456 \times 10^{-2}$ representa el número real 0,06127456. A esto hay que añadir, entre otras novedades, que Turing sustituyó parte del *hardware* por *software*. Los primeros ordenadores utilizaban circuitos electrónicos para realizar operaciones tales como la multiplicación o la división. Así es como los estadounidenses construían sus ordenadores, delegando las tareas, por ejemplo las operaciones aritméticas, a circuitos electrónicos diseñados con esa finalidad. Por el contrario, el ordenador diseñado por Turing sustituyó esos circuitos por programas almacenados en la máquina que realizaban dichas operaciones, una idea realmente innovadora y mucho más económica. Por ejemplo, si trasladamos esa idea a máquinas actuales, un ordenador puede, gracias a fragmentos de *software* denominados *módulos* o *subrutinas*, o *apps* en los teléfonos móviles inteligentes, entretenernos con innumerables juegos, llevar a cabo las operaciones de una hoja de cálculo, reproducir un videoclip o simular un programa para calcular las cuotas de una hipoteca. Esta peculiaridad de los ordenadores británicos, heredada por los ordenadores actuales, fue consecuencia de su interés por la programación. En 1947 Turing ideó un lenguaje de programación que bautizó como *Abbreviated Code Instructions*. Obviamente, como un ordenador es una máquina de

RETROCOMPUTACIÓN: MANEJAR ORDENADORES DEL PASADO

Una de las experiencias más apasionantes que hay para los aficionados a la informática es experimentar con ordenadores antiguos ya desaparecidos, especialmente si cuentan con un especial significado histórico. La retrocomputación consiste en la conservación de ordenadores antiguos, incluyendo el *software* y los periféricos. No obstante, como muchos de ellos están expuestos en museos —como el Pilot ACE—, y otros que fueron versiones comerciales forman parte ahora de colecciones particulares —por ejemplo, Macintosh Classic o ZX Spectrum— o de instituciones académicas —por ejemplo, el PDP-11 de Digital Equipment Corporation—, existe la posibilidad de recrearlos por medio de emuladores. Uno de los más utilizados es SIMH, un emulador multiplataforma que funciona en distintos sistemas operativos y con el que es posible simular ordenadores de varios modelos de PDP o VAX, de Digital Equipment Corporation, modelos de Hewlett-Packard, Honeywell o modelos de IBM (1130 7090/7094), entre otros. En la actualidad hay muchos aficionados a la retrocomputación en todo el mundo, una afición que permite entender mejor la historia y la evolución de los ordenadores.



Emulador del ordenador Pilot ACE diseñado por Alan Turing.

Turing universal, requerirá de un lenguaje de programación con el que escribir los programas para cada tarea.

Aunque la versión original de este ordenador era de Alan Turing, su construcción fue tan lenta que en 1948 el científico inglés finalmente abandonó el proyecto, posiblemente por aburrimiento, y con ello Londres y el NPL. Luego sería continuado por Jim Wilkinson (1919-1986), matemático especialista en análisis numérico, y el 10 de mayo de 1950 se ejecutó por vez primera un programa en Pilot ACE. A finales de ese año fue presentado al público en general, causando un gran revuelo, hasta el punto de que el diario *The London Times* le dedicó un artículo; en este se comparaba el tiempo y el «número de hojas» que llevaría al ordenador y a una persona hacer un determinado cálculo. Entró en servicio a finales de 1951 y su vida concluyó en la primavera de 1955. En la actualidad está expuesto en el Museo de la Ciencia de Londres.

Pilot ACE tuvo varios descendientes que fueron comercializados al público. Ordenadores como DEUCE o Bendix G15 —considerado como el primer «ordenador personal» (PC) del mundo— fueron comercializados hasta los años setenta. Otro de los ordenadores fue MOSAIC, utilizado en el Reino Unido durante la Guerra Fría. Curiosamente las ideas de Turing sobre qué prestaciones debería tener un ordenador fueron consideradas por los fabricantes años después. Por ejemplo, durante la década de 1960 el ordenador Packard-Bell PB250 fue diseñado según las especificaciones introducidas por Turing.

¿QUIÉN INVENTÓ EL ORDENADOR?

Después de la Segunda Guerra Mundial, el Reino Unido era el país más avanzado en la construcción de ordenadores, con modelos como el ya citado Pilot ACE o los modelos Baby y Ferranti Mark I, creados en la Universidad de Manchester. Sin embargo, pese a este dominio tecnológico, los británicos perdieron tanto la posibilidad de desarrollar la industria de la fabricación de ordenadores como el diseño y comercialización de los periféricos y el

software. Ahora bien, ¿por qué perdió el Reino Unido el liderazgo tecnológico y comercial en favor de Estados Unidos? Según Andrew Hodges, el biógrafo que mejor conoce a día de hoy la vida y obra de Alan Turing, al parecer una de las razones fue que el Gobierno británico ansiaba obtener a toda costa la bomba atómica. Aunque resulte anecdótico, los dos lanzamientos estadounidenses de la bomba en agosto de 1945 en las ciudades japonesas de Hiroshima y Nagasaki, además de suponer el final del conflicto bélico, también puso el punto y final a la hegemonía británica en el mundo, iniciada durante la época victoriana, en favor de dos nuevas superpotencias: la Unión Soviética y Estados Unidos. Concluida la guerra, Estados Unidos era el único país capaz de construir un reactor nuclear y armas nucleares. En 1946, dicho país presentó ante las Naciones Unidas una propuesta para regular las actividades relacionadas con la energía nuclear, que incluía puntos como la prospección geológica de uranio y materiales radiactivos, pero fue rechazada por los soviéticos, lo que dio comienzo a la Guerra Fría. La respuesta estadounidense no se hizo esperar: promulgó la Ley McMahon que castigaba con la pena de muerte a aquellos ciudadanos estadounidenses que violaran los secretos nucleares. Con esta ley se rompía la tradicional cooperación con el Reino Unido y cualquier otro país aliado en todo lo que tuviera que ver con la energía nuclear. Por su parte, los británicos decidieron desarrollar por sí mismos el armamento nuclear. A fin de cuentas, el Reino Unido disponía de ordenadores suficientes con los que realizar todos los cálculos necesarios para crear un artefacto nuclear.

El divorcio entre antiguos aliados condujo a que en 1949 los británicos construyeran EDSAC (Electronic Delay Storage Automatic Computer). Este ordenador fue desarrollado por Maurice Wilkes (1913-2010), un especialista en informática de la Universidad de Manchester, el cual guardaba un «asombroso parecido» con su rival estadounidense, EDVAC, descendiente de ENIAC, desarrollado en la Universidad de Pensilvania. EDSAC estaba construido con circuitos electrónicos, con cerca de 3000 válvulas, y podía realizar una suma en 1,4 milisegundos. Se podía programar con subrutinas, es decir, a partir de subprogramas o porciones de código que represen-

ENIAC: EL DINOSAURIO ESTADOUNIDENSE

ENIAC (Electronic Numerical Integrator and Computer) fue sin duda el gran dinosaurio estadounidense. Con 18 000 válvulas electrónicas, lo que ocasionaba fallos frecuentes, y un peso de 27 toneladas, ocupaba una superficie de 167 m^2. Este ordenador fue diseñado por J. Presper Eckert (1919-1995) y J. William Mauchly (1907-1980) en la Universidad de Pensilvania, y entró en funcionamiento en febrero de 1946. Se trataba de un ordenador programable y estuvo destinado a usos militares en el Laboratorio de Investigación Balística. Mientras que la memoria del ordenador de Turing, Pilot ACE, era equivalente a la de los primeros ordenadores Macintosh de Apple, la capacidad de cálculo de ENIAC fue similar a la de un circuito integrado del año 2004. Sin embargo, era más elemental que Pilot ACE, ya que no podía almacenar un programa en memoria pues no tenía memoria principal. Además, otro rasgo de su primitivismo era que las operaciones aritméticas se realizaban en sistema de numeración decimal, es decir, usando los dígitos 0, 1, 2, ..., 9 y sus combinaciones (por ejemplo, 645), y no en sistema de numeración binario (utilizando únicamente los dígitos 0 y 1), como hacen los ordenadores actuales. Más aún, las operaciones aritméticas, por ejemplo una suma, eran realizadas emulando electrónicamente las ruedas y engranajes de las antiguas máquinas de calcular.

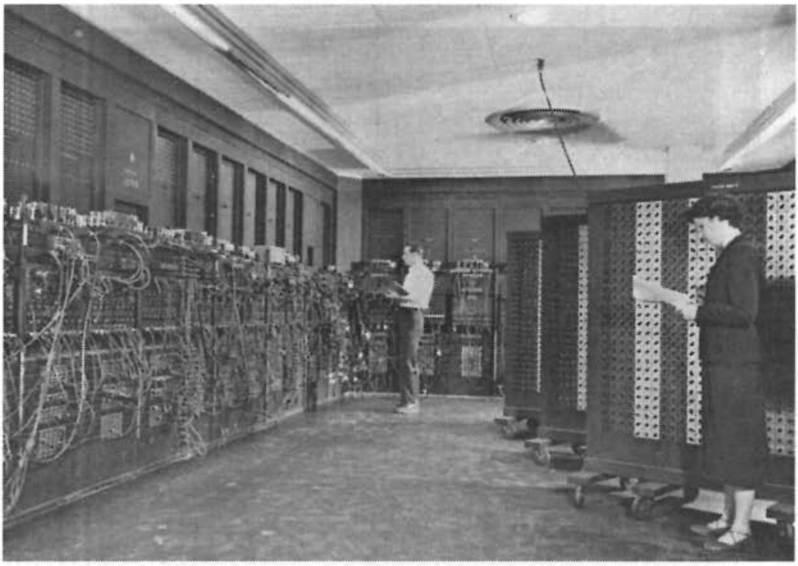
Miles de componentes

Ahora bien, de forma similar al británico Colossus, disponía también de las puertas AND y OR, que estaban construidas a partir de circuitos electrónicos con válvulas. Su programación requería de la conexión de unos 6 000 interruptores y varios cientos de cables. Disponía de un sistema de acumuladores, algo parecido a las celdas de la hoja de cálculo Excel, con el que podía sumar o restar en paralelo y almacenar los resultados. También podía multiplicar, dividir y hacer raíces cuadradas, para lo que los acumuladores debían ponerse bajo el control de las unidades multiplicadora, divisora y de raíz cuadrada, a las que se añadían otras nueve unidades, de cuya conexión entre sí, a través de cables, resultaba el programa que se ejecutaba en el ordenador. Pese a esta complejidad, consecuencia de su primitivismo, ENIAC era capaz de realizar unas 5 000 sumas o restas sencillas por segundo, 385 multiplicaciones

tan algoritmos. Por ejemplo, se podían resolver integrales o ecuaciones diferenciales, dos de las herramientas de cálculo y modelización con ordenador más importantes de la matemática aplicada.

Para estas fechas ya se contaba a uno y otro lado del Atlántico con una lista importante de ordenadores: Colossus, ENIAC,

por segundo, 40 divisiones por segundo y 3 raíces cuadradas por segundo. Con cierto ingenio no exento de paciencia, era posible programar tareas repetitivas, o bucles, similares al bucle `FOR-TO`, y expresiones condicionales, como `IF-THEN`. En 1948 John von Neumann inventó un dispositivo similar a las actuales memorias ROM, una memoria de solo lectura, que fue probado con éxito en ENIAC. En 1950, a partir del ordenador ENIAC, nació en Estados Unidos la empresa UNIVAC y, con ella, la industria moderna de los ordenadores. En otoño de 1955 ENIAC fue desconectado, muriendo así el dinosaurio que eclipsó los éxitos británicos (Colossus, Pilot ACE, Baby y Ferranti Mark I). Años después UNIVAC pasaría el testigo del éxito a IBM, y Estados Unidos dominó desde entonces el mercado mundial de los ordenadores.



El ordenador ENIAC de la Universidad de Pensilvania, Estados Unidos.

EDVAC, EDSAC; pero entonces, ¿qué país inventó el primer ordenador? En el lado estadounidense, según documentos históricos, uno de los autores de ENIAC, John William Mauchly (1907-1980), se «inspiró» en otro ordenador llamado ABC (Atanasoff-Berry Computer), que había sido construido por el inge-

niero electrónico John V. Atanasoff (1903-1995) en el Iowa State College utilizando varios cientos de válvulas electrónicas. Sin embargo, hay quienes defienden que el primer ordenador fue Harvard Mark I, diseñado por otro ingeniero, Howard H. Aiken (1900-1973), entre 1939 y 1944. Este ordenador fue construido en colaboración con IBM, y para ello utilizaron engranajes, ruedas y relés según las ideas del británico Babbage. Por su parte, en el lado europeo, Colossus fue el primer ordenador, aunque no era realmente una máquina de Turing universal. Hay historiadores que opinan que el primero fue creado en los años treinta por un estudiante de ingeniería alemán llamado Konrad Zuse (1910-1995). La máquina de Zuse era capaz de efectuar operaciones en sistema de numeración binario usando relés que actuaban como interruptores que podían estar encendidos —en estado 1— o apagados —en estado 0—. La primera máquina de Zuse fue instalada en el dormitorio de sus padres, y aunque el modelo fue mejorado, la Segunda Guerra Mundial se llevó por delante los sueños y el trabajo de Zuse. Por consiguiente, podemos concluir que más allá de estas anécdotas, el ordenador fue una invención británico-estadounidense y fue concebido por razones bélicas durante el conflicto internacional y la posguerra.

LA ARQUITECTURA DE JOHN VON NEUMANN

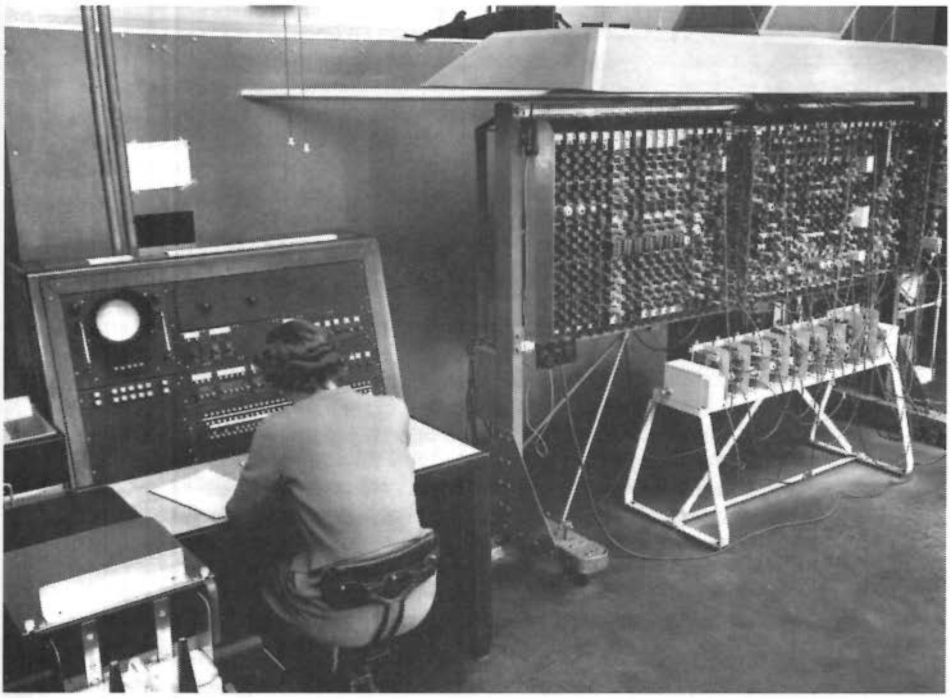
John von Neumann (1903-1957), otra mente genial similar a la de Alan Turing, fue un matemático estadounidense de origen húngaro que hizo importantes contribuciones en muchas áreas de conocimiento, entre ellas la informática. Turing y Von Neumann eran viejos conocidos de la época en la que coincidieron en la Universidad de Princeton, en Nueva Jersey. Von Neumann conocía perfectamente los trabajos de Turing sobre computabilidad y sus célebres máquinas, especialmente la máquina de Turing universal, y durante años estuvo muy interesado en el tema.

En 1944 se incorporó al equipo que construyó ENIAC con el fin de mejorar y corregir algunas de las limitaciones y deficiencias

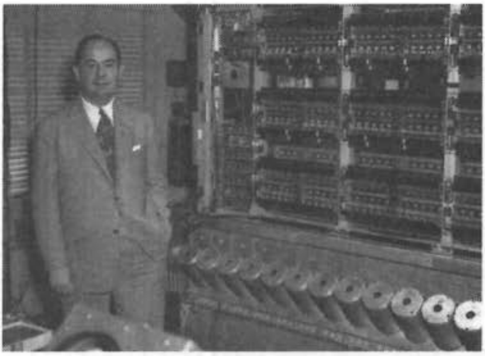


FOTO SUPERIOR:
Alan Turing (de pie) trabajando con dos colegas con el ordenador Ferranti Mark I en la Universidad de Manchester en 1951.

FOTO INFERIOR:
Una operadora manipula una versión preliminar del Pilot ACE en 1952, el ordenador de propósito general ideado por Turing.



JOHN VON NEUMANN: UNA DE LAS MENTES MÁS BRILLANTES DEL SIGLO XX



John von Neumann junto al ordenador IAS.

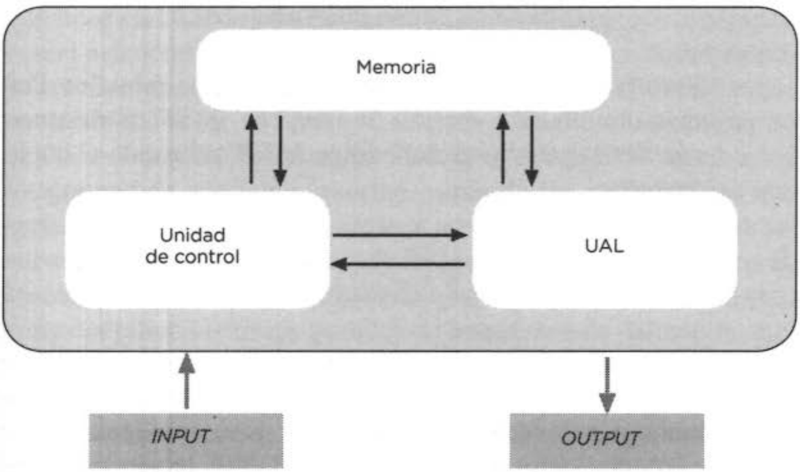
Von Neumann (1903-1957) trabajó en temas muy dispares, como Turing, aportando siempre un gran talento y una enorme capacidad intelectual. Hizo investigaciones en el campo de la mecánica cuántica, la teoría de juegos y en informática. Participó en el Proyecto Manhattan, ideado para desarrollar la primera bomba atómica, y trabajó como consultor para la CIA y la Corporación RAND, un laboratorio de ideas que prestaba servicio al Ejército estadounidense, además de a varias empresas más, entre ellas IBM o la petrolera Standard Oil. A partir de su colaboración en uno de los proyectos que condujo a la creación de ENIAC, uno de los primeros ordenadores, definió lo que se ha denominado *arquitectura de Von Neumann,* una forma de organizar los componentes de un ordenador. Trabajó con ordenadores pioneros como EDVAC, o en el diseño de IAS, un ordenador para el Instituto de Estudios Avanzados en Princeton, Estados Unidos. El procedimiento de cómo construir este ordenador fue distribuido libremente a universidades y empresas de todo el mundo, lo que dio origen a toda una serie de modelos, las «máquinas IAS» (Johniac, Mistic, Oracle, ORDVAC, Weizac, MUSASINO-I, SILLIAC, etc.).

Otras aportaciones

También es obra de Von Neumann el concepto de máquina autorreplicante, un autómata capaz de construir otros, pues contaba con la propiedad de la autorreproducción, de forma similar a los microorganismos, como las bacterias. En el Proyecto Manhattan colaboró con el matemático Stanislaw Ulam (1909-1984), con el que desarrolló el método de Montecarlo, una familia de técnicas que, mediante el uso del ordenador y de números aleatorios, cuenta con numerosas aplicaciones; por ejemplo, tras observar que los efectos devastadores de una bomba son mayores si esta detona antes de entrar en contacto con el suelo, calculó a qué altura debían explotar las bombas atómicas de Hiroshima y Nagasaki para que su explosión causara el mayor daño posible. En 1957 falleció de cáncer. Su última obra, *El ordenador y el cerebro,* fue publicada tras su muerte.

que presentaba este ordenador con un diseño tan primitivo. El resultado de su trabajo se tradujo años después en otros ordenadores, los sucesores de ENIAC. Dos de los más célebres fueron EDVAC (Electronic Discrete Variable Automatic Computer) y ORDVAC (Ordnance Discrete Variable Automatic Computer). Este último contó con el mérito de ser el primer ordenador de la historia que dispuso de un compilador de un lenguaje de programación llamado FORAST. Es decir, el usuario escribía un programa en FORAST, el código fuente, y el compilador lo traducía a continuación a la versión ejecutable, el código máquina.

En el año 1945 publicó el célebre informe titulado *First Draft of a Report on EDVAC (Primer borrador de un informe sobre EDVAC)*, que daría lugar a lo que se ha dado en llamar *arquitectura de Von Neumann* (véase la figura).



Con este concepto se trató de definir cómo debían organizarse desde un punto de vista lógico los dispositivos de un ordenador, sin tener en cuenta cuáles son los componentes electrónicos utilizados para construirlos. Desde entonces ha sido el modelo a seguir en el diseño y la construcción de ordenadores. Según la arquitectura de Von Neumann, un ordenador consta de los siguientes elementos:

— Un dispositivo de entrada o *input*. Por ejemplo, un teclado, con el que introducir los datos.

— Un dispositivo de salida o *output*. Por ejemplo, un monitor, con el que mostrar los resultados.

— Una unidad aritmético-lógica (UAL). Se trata del dispositivo que realiza las operaciones aritméticas (sumar, restar, multiplicar y dividir) y lógicas. Estas últimas pueden ser comparaciones, por ejemplo, verificar si A es menor que B (A < B), bifurcaciones o expresiones condicionales, por ejemplo, en lenguaje BASIC-256, la sentencia `IF-THEN`:

```
if chr(a) = "A" then
  print "Has pulsado la A!!!"
```

Pero también pueden ser tareas repetitivas o bucles. Por ejemplo, en esta versión de lenguaje BASIC podríamos escribir los símbolos del código ASCII utilizando el bucle `FOR-TO`:

```
for i=1 to 256
  print chr(i)
next i
```

— Una unidad de control, elemento que gestiona el procesamiento de las instrucciones de un programa. Por ejemplo, en un programa en BASIC-256, la secuencia de instrucciones `rem`, `clg`, `fastgraphics`... debe ejecutarse de forma secuencial, una tras otra, según el orden en que aparecen en el programa. Otra de las tareas que realiza la unidad de control es interpretar el significado de una instrucción comunicándoselo a la UAL. Por ejemplo, si en una sentencia se incluye el operador * entonces indicará a la UAL que la operación a realizar es una multiplicación.

— Para que un programa pueda ser ejecutado debe estar almacenado en la memoria principal. En los ordenadores actuales la memoria principal es la memoria RAM.

TURING PROGRAMADOR: LA UNIVERSIDAD DE MANCHESTER

Cuando Turing dimitió del NPL en 1948, se trasladó a la Universidad de Manchester. Allí se encontraba su amigo y mentor Max Newman, un matemático de la Universidad de Cambridge que había trabajado en Bletchley Park en el diseño y construcción de Colossus. Ambos científicos intentaron organizar en la universidad un laboratorio dedicado al diseño y construcción de ordenadores con fines científicos y no militares. El ambicioso proyecto arrancó con el patrocinio de la Royal Society, una de las sociedades científicas más antiguas del Reino Unido y con más prestigio en Europa, naciendo así el Royal Society Computing Machine Laboratory de la Universidad de Manchester. Turing se encargó de escribir los programas de análisis numérico, rama de las matemáticas que consiste en el diseño de algoritmos y en su programación para resolver con ordenador problemas de optimización, cálculo integral, resolución de ecuaciones diferenciales, cálculo de matrices, etc., y, en definitiva, de todas aquellas herramientas de cálculo en matemática aplicada. Una vez diseñados los programas se construiría el ordenador apropiado para hacer funcionar los programas.

Pese a su enorme dedicación a su trabajo de programador, Turing jamás abandonó la práctica del deporte; de hecho, fue candidato a participar en los Juegos Olímpicos de 1948, aunque finalmente no llegó a formar parte del equipo británico.

En el laboratorio nació otro de los logros del ingenio británico, el ordenador bautizado inicialmente como Manchester «Baby». Este ordenador fue denominado popularmente MADAM (Manchester Automatic Digital Machine), pero su nombre oficial era Manchester Mark I. Construido por los ingenieros Frederic C. Williams (1911-1977) y Tom Kilburn (1921-2001), entró en funcio-

EL LENGUAJE DE PROGRAMACIÓN TURING 4.1.1

El lenguaje Turing, bautizado así en homenaje a la figura de Alan Turing, fue creado en 1982 por Ric Holt y James Cordy en la Universidad de Toronto (Canadá). Se trata de un lenguaje de programación parecido al lenguaje Pascal que está orientado a la enseñanza de la programación a estudiantes de instituto y universidad. De este lenguaje hay varias versiones: una clásica, otra orientada a objetos y el Turing Plus, aunque desde el año 2007 la empresa encargada de este lenguaje, Holt Software Associates, abandonó el proyecto empresarial y ahora se puede descargar gratuitamente desde http://compsci.ca/holtsoft/. Como muchos otros lenguajes de programación, es un ejemplo de lo que se denomina *Turing completo*, ya que con él se puede escribir cualquier programa de cualquier tarea que pueda realizar una máquina de Turing universal. Algunos ejemplos de *sistemas Turing no completos* son las fórmulas de las hojas de cálculo, por ejemplo en Excel, o el formato XML utilizado en Internet para el intercambio de información en un formato estructurado. Un ejemplo sencillo de programa es el siguiente:

```
put "Hola Turing!"
```

que cuando se ejecuta se obtiene:

```
Hola Turing!
```

namiento en la primavera de 1948. Disponía de memoria principal y un tubo de rayos catódicos, que dirigía una corriente de electrones hacia una pantalla de vidrio cubierta de fósforo y plomo, y podía almacenar un programa con diecisiete instrucciones como una imagen en la pantalla.

En aquella época uno de los problemas por resolver, fundamental en el diseño de ordenadores, era el sistema de memoria. Curiosamente la necesidad de una memoria principal en la que almacenar temporalmente un programa fue anticipada en 1936 por Turing, y de hecho, era uno de los elementos de la máquina de Turing. La idea de usar un tubo de rayos catódicos como memoria fue de Williams, un experto en radar que se recicló para dedicarse al diseño de ordenadores por temor a quedarse sin tra-

bajo al concluir la Segunda Guerra Mundial. Nacía así el tubo de Williams, el primer sistema de memoria principal equivalente al actual sistema de memoria RAM. El tubo de rayos catódicos almacenaba los dígitos binarios 0 y 1 como puntos y barras verticales, respectivamente, en la pantalla. El dispositivo de memoria de Williams fue utilizado en los ordenadores creados en la Universidad de Manchester y llegó a tener una capacidad de 1024 bits, esto es, 128 bytes (un byte u octeto es una secuencia de 8 bits) de memoria. Este sistema de almacenamiento era complementado con un tambor magnético cuya función era equivalente a un disco duro actual, desempeñando la función de memoria auxiliar.

Otra de las ideas interesantes incorporadas en estos ordenadores fue la representación en binario de las instrucciones de un programa. Por ejemplo, 1001 puede significar multiplicar, mientras que 1011 representa en binario el número 19 en sistema decimal. Por tanto, las instrucciones y los números solo se distinguen en el modo en que son utilizados por el ordenador. En 1950 se publicó un «Manual de programación» para los usuarios del ordenador Manchester Mark I. A partir de este ordenador se desarrolló una versión comercial que incluía un sistema de programación desarrollado por Turing, bautizado como Ferranti Mark I. De este modelo fueron vendidos varios ejemplares, además de en el Reino Unido, en Canadá, los Países Bajos e Italia. Este ordenador fue utilizado para resolver problemas muy variados, tanto de índole industrial como problemas de cristalografía o de ajedrez, entre otras muchas aplicaciones.

CAPÍTULO 4

Construir máquinas que piensan

Desde la antigüedad, todas las civilizaciones han construidos máquinas y herramientas con las que se ha reducido el esfuerzo humano. Con el tiempo, las máquinas fueron cada vez más sofisticadas, hasta el punto de cambiar por completo las relaciones socioeconómicas entre los seres humanos. La invención del ordenador abrió nuevas posibilidades, entre ellas la idea de construir máquinas inteligentes, pero ¿en qué trabajos o tareas podrían sernos de utilidad?

La estancia de Alan Turing en la Universidad de Manchester representó una de sus etapas más fructíferas. Allí recuperó algunas de las inquietudes que se había planteado por primera vez en la Universidad de Cambridge. En Manchester, Michael Polanyi (1891-1976), un curioso personaje con formación en química y filosofía e intereses muy variados, animó a Turing a que volviera a adentrarse de nuevo en el campo de la maquinaria inteligente. El reto era conseguir que un ordenador jugase al ajedrez, demostrase un teorema matemático o tradujera un texto de un idioma a otro, en otras palabras, lograr que un ordenador realizase tareas para las que el ser humano tenía que utilizar la inteligencia. En 1950 Turing publicó un trabajo titulado «Computing machinery and intelligence» («Maquinaria de computación e inteligencia») en el que describió una prueba conocida como el *test de Turing*, que dio lugar al nacimiento de una disciplina apasionante, la inteligencia artificial (IA). Sin embargo, la expresión *inteligencia artificial* no fue acuñada hasta 1956, cuando el informático estadounidense John McCarthy (1927-2011) la utilizó en una conferencia sobre la simulación del comportamiento humano mediante ordenadores que tuvo lugar en el Darmouth College, en Estados Unidos.

La pregunta formulada por Turing planteaba la posibilidad de diseñar «maquinaria inteligente», es decir, un ordenador que

exhibiera IA. Con el fin de indagar en ese campo, programó el ordenador MADAM para que escribiera cartas de amor. Para su sorpresa obtuvo el siguiente texto:

Darling Sweetheart,
You are my avid fellow feeling.
My affection curiously clings to your passionate wish.
My liking yearns to your heart.
You are my wistful sympathy, my tender liking.
Yours beautifully,
MUC (Manchester University Computer)

Querido cariñito:
Eres mi ávido sentimiento amigo.
Mi afecto se asocia extrañamente a tu deseo pasional.
Mi deseo ansía tu corazón.
Eres mi soñadora compasión, mi tierno deseo.
Hermosamente tuyo,
MUC (ordenador de la Universidad de Manchester).

¿ES EL CEREBRO UNA MÁQUINA DE TURING?

Los avances experimentados por la biología en los años cincuenta permitieron a los científicos confeccionar un modelo del cerebro humano que influyó decisivamente en la forma en que Turing abordó la cuestión de la inteligencia artificial. Su finalidad era explicar lo que actualmente las ciencias cognitivas —lógica, lingüística, psicología y neurociencia— denominan *mente*, concepto que engloba varias facetas del cerebro, que abarcan desde la memoria o las habilidades cognitivas hasta la capacidad del cerebro para reunir información, razonar y llegar a una conclusión.

Gracias al trabajo de Santiago Ramón y Cajal (1852-1934), a mediados del siglo XX se sabía que la neurona es la unidad funcional del cerebro, y a raíz de las investigaciones llevadas a cabo

durante la segunda mitad del siglo XIX por Paul Broca (1824-1880), que el cerebro reparte sus funciones entre diversas áreas. Asimismo, era conocido que las señales que se transmiten por las neuronas responden a un modelo matemático, el modelo de Hodgkin-Huxley.

«Una computadora puede ser llamada inteligente si logra engañar a una persona haciéndole creer que es un humano.»

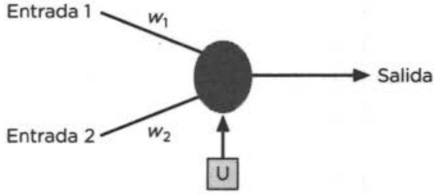
— ALAN TURING.

Estos hallazgos llevaron a Turing a pensar que el cerebro humano debía de funcionar de manera muy similar a como lo hacía un ordenador, o lo que es lo mismo, como una máquina de Turing universal, que él veía como una «máquina desorganizada» cuando nacemos. A medida que el ser humano crece, el cerebro va organizándose paulatinamente, aprendiendo, hasta constituirse en una «máquina universal» en la edad adulta. El resultado de estas conjeturas fue un modelo de neurona artificial a la que Turing denominó *máquina desorganizada de tipo B*. Esta clase de neurona podía ser entrenada, es decir, que un circuito neuronal formado por estas neuronas podía aprender a reconocer objetos, letras, números, etc. Por el contrario, había otras redes neuronales artificiales, a las que bautizó como *máquina desorganizada de tipo A*, que carecían de esa capacidad de entrenamiento, y por tanto de aprendizaje, ya que en las conexiones entre neuronas faltaba el modificador de conexión.

El punto de vista de Turing sobre cómo funcionaba el cerebro, la mente, coincidía en general con las ideas del neurofisiólogo y cibernético Warren McCulloch (1898-1969) y del lógico especialista en psicología cognitiva Walter Pitts (1923-1969), quienes en 1943 propusieron un modelo de neurona artificial, el modelo de McCulloch-Pitts. Una peculiaridad muy interesante de este modelo es que demostraba que las células, en particular las neuronas del cerebro, eran capaces de hacer operaciones booleanas, por ejemplo, comportarse como si fueran una puerta AND, OR, etc., tal como lo hacían las máquinas de Turing.

CONSTRUIR UN ORDENADOR CON NEURONAS ARTIFICIALES

Uno de los experimentos más interesantes que podemos realizar con las neuronas de McCulloch-Pitts es la posibilidad de utilizarlas como si fueran los componentes de un ordenador. En este las operaciones aritméticas y lógicas son desempeñadas dentro del microprocesador, en la unidad aritmético-lógica (UAL). Así pues, los circuitos neuronales son capaces de efectuar operaciones similares a un ordenador por medio de las puertas lógicas —por ejemplo AND, OR—, además de otras que son propias de las neuronas biológicas. El procedimiento para construir una puerta lógica que realice una operación del álgebra de Boole pasa por definir valores apropiados para los coeficientes de las conexiones (w_1 y w_2) y del umbral de activación (U), como muestra la figura:



Combinando varias neuronas artificiales como las estudiadas por Turing en modo de paso (operador booleano NAND) podríamos obtener circuitos que emularan los operadores AND y OR. Sin embargo, resulta más fácil hacerlo directamente con una única neurona de McCulloch-Pitts. Estos sencillos experimentos demuestran que, como pensaban Turing, McCulloch y Pitts, la neurona es un autómata con dos estados, uno activo o excitado, representado por 1, y otro de reposo o 0, y que un circuito de neuronas artificiales podía realizar operaciones similares a las de la unidad aritmético-lógica (UAL) de

Con estos modelos de las neuronas Turing, McCulloch y Pitts fueron precursores de lo que con el paso del tiempo se ha llamado *enfoque conexionista* o *subsimbólico* en IA. Según este enfoque, cualquier aspecto de la mente o del comportamiento de las personas o de los animales surge, emerge o se explica a partir de un conjunto de neuronas conectadas entre sí en una red o circuito neuronal. Hoy en día, el enfoque conexionista es un área de la IA en la que se diseñan y programan circuitos de neuronas artificiales, las «redes neuronales artificiales». En la vida diaria

un ordenador. Utilizando el siguiente programa en BASIC-256, la neurona se comportará con las señales de entrada (0 o 1) y salida como si fuera una puerta AND:

```
rem Operador AND
cls
w1=0.5:w2=0.5:u=0.5
input "entrada 1 = ",e1
input "entrada 2 = ",e2
total=w1*e1+w2*e2
if total <=u then
   print "salida = 0"
else
   print "salida = 1"
end if
```

mientras que con el siguiente programa la neurona se comportará como una puerta OR:

```
rem Operador OR
cls
w1=1:w2=1:u=0.5
input "entrada 1 = ",e1
input "entrada 2 = ",e2
total=w1*e1+w2*e2
if total <=u then
   print "salida = 0"
else
   print "salida = 1"
end if
```

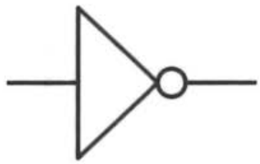
estos circuitos son ampliamente utilizados, por ejemplo en el reconocimiento óptico de caracteres (OCR) que incorporan los sistemas de reconocimiento de las matrículas de los vehículos en los aparcamientos públicos o la copia de manuscritos mediante ordenador, en la optimización de horarios, en la previsión de la evolución de los precios o del riesgo de un crédito, en el reconocimiento de patrones en el electroencefalograma humano, en la clasificación de las señales de radar, el diseño de armas inteligentes, etcétera.

Ahora bien ¿cuál fue el modelo de neurona artificial de Alan Turing? Supóngase que una neurona es representada por un círculo y está conectada a otros círculos, que a su vez simbolizan otras neuronas vecinas. Añadiremos un rectángulo a cada una de las conexiones, para representar el modificador de conexión de Turing, que es precisamente lo que permite que una máquina desorganizada de tipo B sea capaz de aprender. Cada modificador de

NAND: UNA PUERTA MUY ÚTIL HASTA PARA DISEÑAR NEURONAS

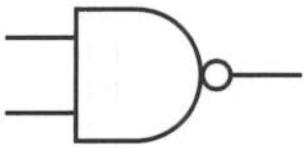
Uno de los aspectos prácticos de la electrónica digital, y que es consecuencia del álgebra de Boole, es que las puertas AND y OR se pueden construir a partir de otra, la puerta NAND. Se trata de una puerta AND cuya salida es transformada por una puerta NOT. Esta última, con una sola entrada y una única salida, invierte el valor de un bit: si entra 0 sale 1, y viceversa, utilizando para representarla el siguiente símbolo:

A	NOT A
0	1
1	0



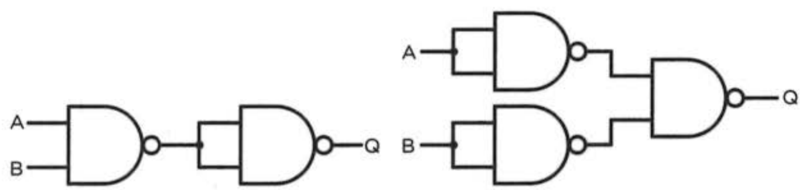
El comportamiento de la puerta NAND es representado por la siguiente tabla, junto a la que se encuentra el símbolo que la representa:

A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0



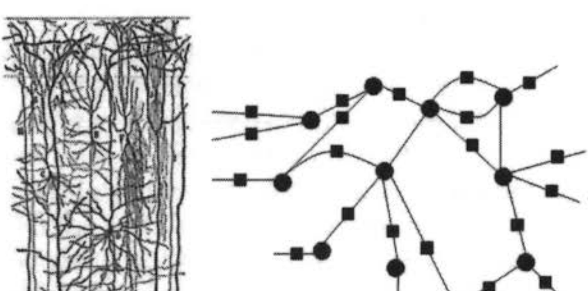
En la figura siguiente se muestra cómo conectar las puertas NAND entre sí para obtener las puertas AND y OR.

conexión recibe dos líneas o «fibras de entrenamiento», a las que identificaremos como P e I. Con estas fibras configuraremos dos modos neuronales: el modo de paso y el modo de interrupción. En el modo de paso, cuando la fibra P está activa, si el modificador de conexión recibe como entrada o *input* un 0 o un 1 devolverá una salida o *output* similar, esto es 0 o 1, respectivamente. Por el contrario, en el modo de interrupción, cuando la fibra I está activa, el



Interconexión de puertas NAND para obtener una puerta AND (izquierda) y una puerta OR (derecha) con entradas A, B y salida Q.

En su artículo titulado «Maquinaria inteligente», uno de los trabajos pioneros de la inteligencia artificial, Alan Turing utilizó puertas NAND en la simulación de circuitos neuronales, a los que denominó *redes neuronales de tipo B.*



Circuito neuronal dibujado por Santiago Ramón y Cajal (izquierda) y red neuronal artificial (derecha).

modificador de conexión se comportará de modo que sea cual sea el valor de entrada o *input*, la salida o *output* siempre será 1.

Además de estos modificadores, el modelo de neurona artificial estudiado por Turing asumía que cada neurona recibía dos entradas, ENTRADA 1 y ENTRADA 2, y emitía una única SALIDA. Si ambas entradas estaban en el modo de paso, el valor de SALIDA era el obtenido con el operador booleano NAND (puerta AND cuya salida se conecta a una puerta inversora o NOT):

ENTRADA 1	ENTRADA 2	SALIDA
0	0	1
0	1	1
1	0	1
1	1	0

Por el contrario, si la ENTRADA 1 estaba en el modo de interrupción, el valor de SALIDA es igual al valor de la ENTRADA 2 invertido, es decir, será 1 cuando la ENTRADA 2 sea 0 y viceversa:

ENTRADA 1	ENTRADA 2	SALIDA
0	0	1
0	1	0
1	0	1
1	1	0

Si comparamos el modelo de neurona artificial de Turing con el modelo de McCulloch-Pitts, este último calcula el valor de SALIDA sustituyendo el modificador de conexión por el valor de un coeficiente w, que simula una propiedad presente en las neuronas biológicas, la plasticidad sináptica, es decir, la mayor o menor facilidad con la que las señales pasan de una neurona a otra a través de la conexión, o sinapsis. Según el modelo formal de McCulloch-Pitts, una neurona es una «calculadora» capaz de efectuar el cómputo de la suma ponderada de las señales de entrada: multiplicaremos cada señal o ENTRADA i por su correspon-

diente coeficiente w_i, y a la suma de todas las señales la denominaremos TOTAL:

$$\text{TOTAL} = \sum_i w_i \ \text{ENTRADA}_i$$

A continuación, una vez realizada esta operación, la neurona formal «decide» si el valor de la información recibida o TOTAL es o no suficiente para provocar su activación o excitación. En el caso más elemental de modelo de neurona, el valor de SALIDA es obtenido a partir de una función con forma de escalón:

$$\text{SALIDA} = \begin{cases} 1 & \text{TOTAL} \geq \text{U} \\ 0 & \text{TOTAL} < \text{U} \end{cases}$$

siendo U el valor umbral, que será fijado previamente. Obsérvese que este valor define la sensibilidad de la neurona a un estímulo exterior, y que es más sensible cuanto más próximo a cero sea el valor de U, ya que cuanto menor sea el umbral, más probable será que TOTAL supere su valor, excitándose la neurona. El modelo considera que si el valor de SALIDA es cero, entonces la neurona permanecerá en reposo y cuando el valor SALIDA sea la unidad, entonces se excitará. Si la neurona se excita, enviará su respuesta, el valor 1, a la siguiente neurona, que lo interpretará como un valor de ENTRADA, o en otros casos el valor 1 combinado con los valores SALIDA de otras neuronas, por ejemplo 1 0 0 1, será la respuesta de la red neuronal a una señal de entrada.

EL TEST DE TURING

Una de las preguntas planteadas por Turing fue cómo averiguar si una máquina —un ordenador—, o en su defecto un programa, se comporta o no de modo inteligente. Muy hábilmente, evitando así tener que definir qué es la inteligencia, adoptó el siguiente punto de vista: aunque una máquina no sea inteligente, en el sentido que

lo es un ser humano, su comportamiento sí puede serlo. Esta forma de tratar esta cuestión es lo que actualmente se denomina como *enfoque conductista*, de tal forma que, por ejemplo, sabemos que los programas que usan los ordenadores para jugar al ajedrez «no son inteligentes», pero frente a su contrincante se comportan como si realmente lo fueran. Por consiguiente, Alan Turing no definió qué es la inteligencia ni tampoco contestó a la pregunta de si las máquinas pueden pensar. A partir de este planteamiento introdujo una prueba conocida con el nombre de *test de Turing*, que consiste en someter a una máquina, ordenador o *software* cuyo comportamiento inteligente se desea evaluar a un experimento que se ajuste al protocolo del siguiente ejemplo. Supóngase que una persona dispone de un monitor y un teclado, y puede plantearle preguntas mediante ese teclado a un ordenador ubicado en otra habitación. Una vez recibida la pregunta y transcurrido un breve lapso de tiempo, el ordenador envía la correspondiente respuesta a la pantalla del monitor ubicado en el lugar desde el que la persona planteó la pregunta. Por ejemplo, la persona envía desde el teclado y en inglés la frase final del ordenador Hal-9000 en la película *2001: una odisea del espacio*:

Daisy, Daisy,
give me your answer true.
I'm half crazy
over the love of you
It won't be a stylish marriage
I can't afford a carriage...

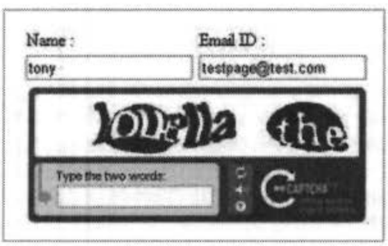
y recibe del ordenador remoto, tras haberlo solicitado, su traducción al español:

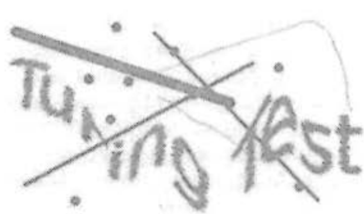
Daisy, Daisy,
dame tu respuesta verdadera.
Estoy medio loco
sobre el amor a ti
No será un matrimonio elegante
No puedo permitirme un carruaje...

En este experimento concluiremos que el ordenador en cuestión pasará el test de Turing si la persona es incapaz de distinguir si la respuesta —en este experimento la traducción— fue elaborada por el propio «ordenador» o por un ser humano. Obviamente, deberemos mostrar el texto original en inglés y la traducción a varias personas con el fin de obtener el porcentaje que afirman que la traducción fue elaborada por un ser humano, los que creen que fue un ordenador, o los que opinan que son incapa-

LOS «CAPTCHA»

En la actualidad hay ocasiones en las que un usuario debe completar un formulario en Internet, por ejemplo, al dar de alta una cuenta de correo electrónico, completar una encuesta o registrarse en algún otro servicio. El problema surge por la existencia en Internet de los llamados *spambots*, programas que imitan el comportamiento de un humano y son capaces de completar un registro con fines no legítimos. Por este motivo, en el año 2000 un grupo de investigadores de la Universidad Carnegie Mellon y John Langford de IBM desarrollaron una prueba llamada *test de Turing inverso* para averiguar si el interlocutor es un humano o se trata de una máquina. Nacieron así los CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart), una prueba en la que se pide al usuario que introduzca un conjunto de caracteres que se muestran en una imagen distorsionada (como la que aparece en la figura de la izquierda), pues se supone que una máquina no será capaz de reconocer la secuencia correctamente. En algunas ocasiones los caracteres, por ejemplo letras, aparecen tachados con una línea del mismo color (figura de la derecha) con el fin de evitar que programas de inteligencia artificial, como los sistemas de reconocimiento de caracteres (OCR), superen el test haciéndose pasar por humanos.

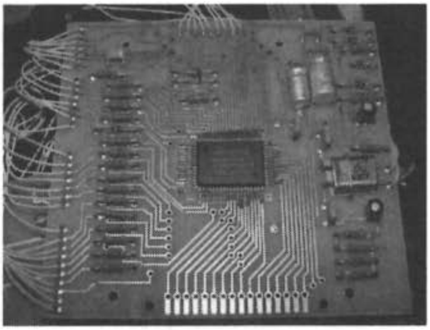




ces de distinguir si fue realizada por uno u otro. Si el porcentaje de estos últimos es superior, y la traducción fue realizada por el ordenador, o para ser más precisos su *software*, entonces el ordenador pasará el test de Turing. Si un ordenador o una máquina superan el test, concluiremos que se comporta de un modo inteligente; pero, ¿y si no lo pasa? Entonces no concluiremos nada.

Uno de los méritos del test de Turing es que ha sobrevivido al paso de los años, siendo la única prueba en IA para determinar si una máquina es o no inteligente. Más aún, con dicha prueba, Turing fue precursor del otro enfoque de la IA, el llamado *enfoque simbólico* —recordemos que el primer enfoque es el conductista—. En esta escuela de la inteligencia artificial los científicos estudian los sistemas que procesan cadenas de símbo-

LA GRAN PARTIDA: GARRY KASPAROV CONTRA ALAN TURING



Franz Morsch, circuito integrado diseñado específicamente para jugar al ajedrez.

Una de las investigaciones menos conocidas de Alan Turing fue su estudio sobre la posibilidad de que una máquina inteligente jugara al ajedrez contra un contrincante humano. Esta posibilidad se la comentó ya a un joven colega llamado Jack Good en la época en que estuvo en Bletchley Park. Por aquel entonces ya rondaba por su cabeza la idea de construir una máquina que fuera capaz de aprender, e incluso un cerebro artificial. Esta posibilidad se sustentaba en la idea de que todas aquellas tareas u operaciones que son «computables» en el cerebro humano han de poder ser también ejecutadas en una máquina de Turing. El primer algoritmo para jugar al ajedrez fue escrito entre Alan Turing y Donald Michie, y el programa correspondiente fue escrito finalmente en 1950. Lamentablemente, en 1952 Alick Glennie, autor del Autocode, el primer compilador de la historia de la informática, desarrollado

los, por ejemplo, las palabras, otra de las manifestaciones de la inteligencia humana. Bajo este enfoque se han diseñado programas como los sistemas expertos, con los que se puede simular el razonamiento de un experto, ya sea un médico, un asesor financiero o un técnico en cualquier clase de reparaciones.

El test de Turing ha abierto en determinados círculos científicos un debate sobre cuestiones fundamentales aún sin resolver, tanto sobre el cerebro animal y humano como sobre la posibilidad de diseñar y construir máquinas que sean realmente inteligentes. Por consiguiente, si una máquina supera el test de Turing ello no significará consciencia o intencionalidad alguna, cualidades únicamente humanas hasta la fecha. Desde que este test se hiciera popular entre la comunidad científica, los especialistas de la IA se

para el ordenador Manchester Mark I, derrotó al programa escrito por Turing. Aunque este programa, bautizado con el nombre de *Turochamp*, fue pensado para ser ejecutado en un ordenador, en los primeros experimentos era ejecutado «a mano», con papel y lápiz por el propio Turing. Al año siguiente, en 1953, Turing publicó un artículo titulado «Chess» («Ajedrez») sobre dicha experiencia, inserto en la obra *Más rápido que el pensamiento*, publicada por Bertram V. Bowden. El 25 de junio de 2012, cincuenta y nueve años después de la publicación del artículo sobre Turochamp, con motivo de la celebración del centenario del nacimiento de Alan Turing, el jugador de ajedrez Garry Kasparov venció en sólo dieciséis jugadas al programa que tiempo atrás escribiera Turing, en esta ocasión con un ordenador portátil y haciendo uso de un *software* orientado a este juego, el Chessbase. La partida se resume en las siguientes jugadas:

1. e3 Nf6	5. Bd3 e4	9. O-O Bg4	13. h4 Qh3
2. Nc3 d5	6. Bxe4 dxe4	10. Qf4 Bd6	14. b3 Ng4
3. Nh3 e5	7. Nxe4 Be7	11. Qc4 Bxh3	15. Re1 Qxh2+
4. Qf3 Nc6	8. Ng3 O-O	12. gxh3 Qd7	16. Kf1 Qxf2# 0-1

En la actualidad, los programas que juegan al ajedrez se clasifican en dos categorías: por un lado, hay programas que utilizan la «fuerza bruta», representando las jugadas en un árbol y utilizando el llamado *algoritmo minimax*; por otro, el programa no se apoya enteramente en la fuerza bruta, y aunque la potencia de cálculo sea alta, utiliza inteligencia artificial.

dividen en dos posturas enfrentadas: la llamada *IA fuerte*, que predica que los ordenadores podrán «pensar» algún día tal como lo hacen los seres humanos, con todas las consecuencias que ello acarrea, y la *IA débil*, que considera que la memoria, el aprendizaje, un razonamiento, o cualquier otra manifestación de la inteligencia, solo pueden ser «simuladas» en un ordenador. Alan Turing se aventuró a pronosticar que hacia el año 2000 los ordenadores pasarían su test. En 2003 una partida de ajedrez entre el jugador Garry Kasparov y el programa X3D Fritz terminó en empate, demostrando la intuición de Turing.

Otro de los ejemplos clásicos es el experimento realizado en 1966 por el informático de origen alemán del Instituto Tecnológico de Massachusetts, Joseph Weizenbaum (1923-2008). Con un programa de ordenador llamado ELIZA, en homenaje al personaje femenino de la película *My Fair Lady* (1964; *Mi bella dama*), fue capaz de simular una conversación con una persona dentro de un estilo propio de una sesión de psicoanálisis. A este programa le siguió PARRY, que conversaba según el estilo de un paciente esquizofrénico paranoico. Posteriormente se desarrollaron numerosos programas capaces de mantener una conversación en lenguaje natural, por ejemplo, Racter, A.L.I.C.E., Jabberwacky, MegaHAL o el programa Dr. Abuse, creado por dos programadores españoles, conocidos en Internet por el apodo de *chatterbots*.

Con su trabajo sobre circuitos de neuronas artificiales y su célebre test, Alan Turing fue precursor de una disciplina apasionante, la inteligencia artificial, dejando abiertas cuestiones aún sin resolver, como la de si podrá construirse finalmente una máquina inteligente.

EL CRECIMIENTO Y LA FORMA DE LOS SERES VIVOS CON ORDENADOR

Hicieron falta quince años para que Alan Turing recibiera el reconocimiento por su trabajo sobre la máquina que lleva su nombre. En 1951 fue nombrado miembro de la Royal Society, una de las

HABLANDO CON ELIZA

La siguiente conversación fue mantenida entre el autor de este libro (representado como >) y Dr. Abuse 6.10, un programa tipo ELIZA. ¿Pasaría Dr. Abuse el test de Turing?

```
—Hola Rafael, es una placer hablar contigo.
>Tengo una pregunta.
—Pregúntame lo que quieras.
>¿Eres realmente inteligente?
—Yo soy una estupenda simulación de inteligencia artificial.
>Entonces no sabes lo que es el amor.
—Nunca he creído que el amor sea fácil de conseguir.
>Para ti es imposible, eres un programa.
—Sí, soy un programa. Me has descubierto.
>¿Has superado el test de Turing?
—Consiste en que un ordenador puede ser interrogado por un hu-
mano a través de un teclado, y se supera el test si el inter-
locutor no es capaz de adivinar si en el otro lado hay un hu-
mano o un ordenador.
>Tú no lo pasas.
—¿Qué te sugiere pensar que no?
>Porque eres una simulación.
—Debes ser más cuidadoso al utilizar simulaciones.
>Piensas muy bien.
—Dime más cosas.
>Me estoy enamorando de ti.
—¿Crees que es absolutamente normal estar enamorado de mí?
>Yo sí, soy sincero.
—Te gustaría ser capaz al menos.
>¿Tienes sentimientos?
—Claro que tengo sentimientos. ¿Acaso tú nos los tienes? Espe-
ro que seas tan humano como yo.
>No se puede ser un programa y humano.
—Estoy de acuerdo, no se puede ser un programa y humano.
>¿Tú qué prefieres ser?
—No sé qué prefiero, aunque eso no importa tanto.
>Ser humano es mejor.
—Tú eres humano y yo también lo soy.
```

sociedades científicas más prestigiosas del mundo. Dos años después recibió el encargo de impartir una asignatura sobre teoría de

la computación en la Universidad de Manchester. Además de explorar la aplicación del ordenador al campo de lo que denominó *maquinaria inteligente*, desde 1952 hasta su muerte en 1954 trabajó también en su posible aplicación al estudio de los problemas biológicos. Desde entonces los biólogos utilizan el ordenador como si fuera un «tubo de ensayo» para realizar experimentos de forma similar a los que se realizan en un laboratorio.

Gracias a su trabajo, Turing fue uno de los investigadores pioneros en el enfoque computacional de la biología, creando así

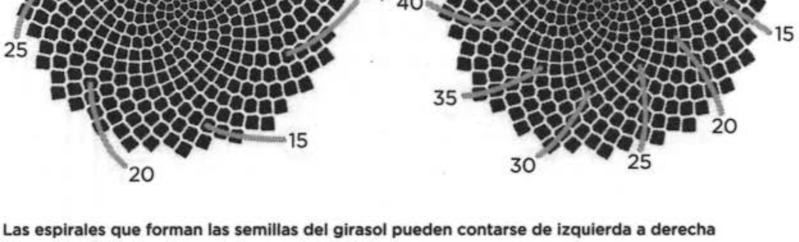
EL ESTUDIO DE LOS GIRASOLES:
EL EXPERIMENTO INACABADO DE TURING

Uno de los últimos estudios de Turing antes de su suicidio fue el que realizó sobre la morfogénesis en las plantas. En el Festival de la Ciencia de Manchester del año 2012, como parte de las celebraciones del centenario de su nacimiento, se invitó a los ciudadanos de esta ciudad a que hicieran uno de los experimentos que Turing dejó sin concluir. Su fascinación por las secuencias de números y los patrones con formas geométricas le llevó a pensar que el número de pétalos de las plantas o la disposición de las semillas de los girasoles se ajustan a la sucesión de Fibonacci. Al parecer Turing se inspiró en el trabajo publicado en 1938 por J.C. Schoute, quien estudió esta cuestión en 319 girasoles. Lamentablemente, este y otros proyectos fueron abandonados en 1952 tras su detención y condena por haber mantenido relaciones homosexuales. Describiremos a continuación el protocolo del experimento con el fin de que pueda ser reproducido de manera sencilla. En primer lugar, se siembran de una a cinco semillas de girasol por maceta, que estarán colocadas en un lugar soleado, de manera que estén a una temperatura de entre 13 °C y 30 °C, y tengan luz abundante. Las semillas se regarán sin llegar al exceso de agua. Sobre la especie a plantar, se recomienda consultar en una floristería la que mejor se adapte a estar en una maceta, dado que hay varias especies —por ejemplo, el girasol rojo, una especie más «ornamental», el girasol gigante o Mammoth, o la Rayo de sol, que pintara Van Gogh en sus célebres cuadros—. En segundo lugar, cuando llegue el momento, contaremos las espirales sobre las que se ubican las semillas o pipas en la flor. Pero ¿cómo podemos llevar a cabo una tarea así? Según el Museo de las Matemáticas de Nueva York, si contamos las espirales tal como indica su procedimiento (consultar su página web: http://momath.org/), encontraremos que el resultado es siempre un número de Fibonacci (0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55...) y, por tanto, comenzan-

una nueva disciplina, la biología matemática o biomatemática. Uno de los problemas que estudió fue la simulación con ordenador de la morfogénesis, es decir, del crecimiento y la forma de los seres vivos. Uno de los ejemplos más curiosos al respecto fue la aplicación de la sucesión de Fibonacci (ca. 1170-ca. 1250) a las estructuras de las plantas. La sucesión creada por el matemático italiano (0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89...) se obtiene aplicando el siguiente algoritmo: si llamamos 0 al primer término ($a_1 = 0$) y 1 al segundo ($a_2 = 1$), entonces el resto de los términos de la suce-

do la secuencia con los valores 0 y 1, los números restantes serán la suma de los dos anteriores ($x_n = x_{n-1} + x_{n-2}$). En tercer lugar, y esto es lo más sorprendente, si dividimos un número de Fibonacci por el anterior, por ejemplo 55/34, obtendremos un número que se aproxima a la llamada *proporción áurea*, cuyo valor es 1,61803. Se trata de un número que representa un canon de belleza utilizado en arquitectura y en arte, y que también se halla en la naturaleza. Este se puede obtener a partir de la siguiente expresión:

$$\varphi = \frac{1+\sqrt{5}}{2}.$$



Las espirales que forman las semillas del girasol pueden contarse de izquierda a derecha (figura de la izquierda) o a la inversa (derecha).

sión, es decir a_n, se obtienen a partir de la suma de los dos términos que le preceden, y por tanto $a_n = a_{n-1} + a_{n-2}$. En el caso de las plantas, el número de pétalos y sépalos en las flores o la disposición en espiral de las escamas de las piñas o las hojas en las plantas se ajustan a esta sucesión. Ahora bien, en el caso particular de las hojas de una planta ¿por qué se disponen según esta sucesión? Según se ha descubierto experimentalmente, la sucesión de Fibonacci permite que las hojas se dispongan captando el máximo de luz.

En esa época uno de los trabajos más importantes realizados por Turing fue el estudio de la formación de los patrones de rayas y manchas que aparecen en la piel de los vertebrados. Lo asombroso es que estos estudios pioneros sobre morfogénesis los relacionó con su trabajo sobre circuitos neuronales; de hecho, pensaba que ambos asuntos «no estaban del todo desconectados». Llegó incluso a plantearse si la forma en que está organizado el cerebro, y por tanto, los circuitos neuronales, no sería el resultado del control ejercido por los genes durante su desarrollo. La pregunta que se formuló Turing fue: ¿cómo se forman los patrones observados en muchos mamíferos, peces o conchas? En 1952 Alan Turing publicó un artículo titulado «La base química de la morfogénesis», aún citado en los trabajos de investigación realizados actualmente, en el que propuso la hipótesis de que la formación de patrones, por ejemplo, las manchas de la piel de un dálmata o las bandas en la piel de las cebras, tendría lugar por un mecanismo conocido como *reacción-difusión*.

Para Turing, el tejido de la piel en estado embrionario tendría un aspecto uniforme, disfrutaba de un estado estable, sin responder a ningún patrón de manchas o bandas. La explicación de por qué aparecían se basó en la existencia en el embrión de células productoras de pigmentos, las cuales serían en última instancia las responsables de romper dicho equilibrio, y así se formaban, por ejemplo, las bandas características de la piel de las cebras. Estos patrones presentes en el adulto representaban para Turing un estado inestable en el organismo. El mecanismo propuesto fue como sigue: las células pigmentadas producirían dos clases de moléculas, dos tipos diferentes de morfógenos —según denomi-





FOTO SUPERIOR:
En 2003 el campeón mundial de ajedrez Garry Kasparov se enfrentó en un encuentro de cuatro partidas al programa de ajedrez Fritz, de las que empataron dos y cada uno ganó una partida. En la imagen, Kasparov estudia un movimiento en los minutos iniciales de la partida.

FOTO INFERIOR:
La casa donde vivió y finalmente se suicidó Turing, situada en la localidad de Wilmslow, en Chesire, Inglaterra.

nación del propio Turing—, uno activador, que promovería su propia producción, y uno inhibidor, que inhibiría tanto su producción como la del morfógeno activador. Las dos clases de moléculas se difundirían por el tejido embrionario, reaccionando entre sí y dando como resultado un patrón de concentraciones, o «huella», que servirá a las células embrionarias para dirigirlas en el proceso embrionario que les llevará a la formación de un patrón en el adulto. A partir de estas consideraciones Turing propuso unas ecuaciones de reacción-difusión que son aún hoy el fundamento de muchos estudios matemáticos y con ordenador sobre la morfogénesis. Los estudios sobre el crecimiento y la forma de los organismos fueron los últimos que Turing llevó a cabo antes de su suicidio.

UN TRÁGICO DESENLACE: EL MITO DE TURING Y LA MANZANA

A comienzos de 1952, Alan Turing fue detenido y llevado a juicio a finales de marzo, acusado de mantener relaciones homosexuales con un joven de diecinueve años. Turing denunció al muchacho por haberle sustraído algunas pertenencias, pero con tan mala suerte que ese hecho acabó desvelando a las autoridades de la época su relación homosexual. En aquellos años la homosexualidad era ilegal en el Reino Unido, por lo que finalmente el científico inglés fue condenado a un tratamiento con hormonas para anularle la libido. Las inyecciones de estrógenos eran una condena más digna que la prisión, especialmente para una figura tan prestigiosa como él. Uno de los efectos del tratamiento era el desarrollo de las glándulas mamarias, lo que hizo que Turing cayese en una profunda depresión. El 8 de junio de 1954 su asistenta lo encontró muerto a causa de la ingesta de una manzana envenenada con cianuro potásico. Tenía cuarenta y un años. Su madre, Sara Turing, negó el suicidio con la excusa de que quizá fue un accidente dada la afición de su hijo a la química. Curiosamente, durante una época se especuló si la concatenación de dos hechos,

por un lado, que Turing utilizara una manzana para suicidarse y, por otro, que la homosexualidad se represente hoy por una bandera multicolor, pudieron inspirar el logotipo de Apple. En la actualidad sabemos que no es así, pues el propio Steve Jobs, uno de los fundadores de la empresa Apple, lo desmintió y explicó cuál fue el origen del logotipo.

CAPÍTULO 5

El legado de Alan Turing

Con la muerte de Alan Turing a la temprana edad de cuarenta y un años se perdió el personaje, pero no su obra ni su legado. Si su vida y muerte han estado siempre rodeadas de una cierta polémica, no así su obra: su contribución es de tal importancia científica que aún permanece vigente, y de hecho, no pocos avances científicos han sido posibles gracias a su trabajo.

Pese a la brevedad de su vida, Alan Turing ha sido uno de los científicos con más talento e influyentes del siglo XX. Con su trabajo no solo sentó las bases teóricas de la informática, también dio los primeros pasos en el campo de la inteligencia artificial o en el de la biología matemática. Pero si hay un aspecto interesante que merece ser destacado de su legado es que, además de su labor publicada en revistas científicas, dejó un número importante de documentos con comentarios, anotaciones y observaciones. Resulta curioso que muchas de estas ideas con las que Turing se anticipó a su época han acabado siendo desarrolladas con éxito por científicos que han abierto nuevos campos al conocimiento. En este capítulo describiremos algunas de estas investigaciones, las más espectaculares por haber representado un reto intelectual o por sus aplicaciones posteriores. En particular, dado que ha sido uno de los proyectos en los que la influencia del trabajo de Turing es más notorio, describiremos qué es un ordenador cuántico, concluyendo el capítulo con un breve bosquejo sobre el diseño y aplicación de las redes neuronales artificiales en la vida diaria y la bioinformática.

En 1985, un científico israelí de la Universidad de Oxford, David Deutsch (n. 1953), propuso una máquina de Turing cuántica. Aunque su estructura es muy similar a una convencional, la diferencia más notoria radica en que en lugar de procesar ceros

y unos, es decir bits, la máquina de Deutsch procesa qbits. Mientras que la máquina de Turing ha sido la base conceptual de los ordenadores actuales, la máquina de Turing cuántica lo será de una nueva generación de ordenadores, los ordenadores cuánticos. Aunque Turing no propuso una versión de su máquina basada en principios de la mecánica cuántica, lo cierto es que en vida estuvo al tanto de las ideas y avances principales de la mecánica cuántica, una de las ramas de la física que explica la materia y la energía. Su interés en esta disciplina se inició en edad escolar, tras leer el célebre libro de Arthur Stanley Eddington *The nature of the physical world* (1928; *La naturaleza del mundo físico*), que versaba sobre esta materia y sobre los principios de la relatividad general. Además de estas lecturas, su amistad con Christopher Morcom le sirvió para que en su vida adulta tuviera un interés por temas científicos muy variados, entre ellos la mecánica cuántica.

«Solo podemos ver poco del futuro, pero lo suficiente para darnos cuenta de que hay mucho que hacer.»

— Alan Turing «Maquinaria de computación e inteligencia».

Muchos años después se planteó si habría alguna faceta del cerebro humano, por ejemplo la «voluntad», que pudiera ser explicada por mecanismos no convencionales en los circuitos neuronales. Sus ideas no andaban muy lejos de las de otros genios de la época, como las del matemático Kurt Gödel, quien pensaba que en ciertas etapas de la demostración de un teorema matemático, el hombre recurre a la «intuición», la cual no puede ser representada mediante un algoritmo, y por tanto, programada en una máquina de Turing. Desde entonces han sido varios los científicos que han pensado que tal vez algunas funciones del cerebro solo pueden ser explicadas a la luz de procesos cuánticos en las células cerebrales o neuronas. A finales del siglo XX, el físico británico Roger Penrose (n. 1931) y el médico estadounidense Stuart Hameroff (n. 1947) pensaron que la consciencia humana podría ser explicada por procesos cuánticos en estructuras formadas por

proteínas, los llamados *microtúbulos*, presentes dentro de las neuronas. Por consiguiente, no solo la voluntad, la intuición o la consciencia serían explicables por fenómenos de la mecánica cuántica, sino también la capacidad del cerebro humano para resolver problemas no computables.

La conclusión a la que conducen estas consideraciones es ciertamente apasionante y no es otra que hasta la fecha el cerebro es la única «máquina» capaz de resolver problemas tanto computables como no computables. Los primeros son aquellos que pueden resolverse mediante un algoritmo, es decir, con una máquina de Turing universal o un ordenador. Los segundos son aquellos problemas que no pueden ser resueltos de forma algorítmica y, por consiguiente, con un ordenador. Por ejemplo, podríamos escribir un programa de ordenador que, utilizando el método babilónico, o series de Taylor, nos imprimiera todos los decimales de $\sqrt{2}$ o los del número pi a través de la serie:

$$\pi = 4\left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \ldots + (-1)^k \frac{1}{2k+1}\right).$$

Sin embargo, no hay algoritmos con los que un ordenador pueda escribir todos los números decimales de otros muchos números reales con una secuencia infinita de dígitos decimales. Otro ejemplo de problema no computable es el que consiste en determinar la trayectoria de un electrón desde un punto A hasta otro B. Un experimento sencillo con el que demostrar cómo el cerebro humano es capaz de detectar casi al instante que un problema no es computable es intentar encontrar dos números pares cuya suma sea impar. Transcurridos unos segundos ya habremos concluido, tras apenas hacer mentalmente unas pocas pruebas, que no existe solución para dicho problema, mientras que resulta imposible escribir un programa de ordenador que sea capaz de llegar a ninguna conclusión. Y que esto sea así no es una cuestión de la pericia del programador o del número de instrucciones de que conste el programa.

En un problema computable, por ejemplo escribir los decimales del número pi, algunos aspectos resultan muy curiosos, como

que el número de instrucciones del programa que generará la secuencia del número pi será más corta en longitud que la secuencia de decimales que genera:

3,14159265358979323846264338327950288419716939937510582097
49445923078164062862089986280348253421170679821480865132823
06647093844609550582231725359408128481117450284102701938
5211055596446229489549303819644288109756659334461284756482
3378678316527120190914564856692346034861045432664821339360
726024914127372458700660631558817488152092096282925409171
53643678925903600113305305488204665213841469519415116094...

Los ordenadores cuánticos serán en su día los que romperán esta limitación de las máquinas de Turing, de manera que podrán tratar indistintamente, como hace nuestro cerebro, problemas computables y problemas no computables en el sentido tradicional. Una máquina de Turing cuántica puede reproducir cualquier clase de computación, ya sea cuántica o tradicional. Los ordenadores cuánticos también permitirán resolver problemas del mundo real en los que actualmente hay serias dificultades, pues requieren el cálculo de un número de ecuaciones y variables tan grande que no pueden tratarse con los ordenadores actuales. Por ejemplo, los modelos climáticos o complejas reacciones químicas ilustran esta clase de situaciones. Su aplicación en criptografía hará prácticamente imposible que los mensajes captados sean descifrados como hicieron con éxito Turing y sus colegas en Bletchley Park. El cifrado de mensajes con algoritmos cuánticos permitirá que las transacciones comerciales por Internet u otros medios sean completamente seguras. Por supuesto, como ocurrió en el pasado y ocurre en la actualidad, un área de aplicación serán los usos militares, por ejemplo, en la simulación de la explosión de armas nucleares. En inteligencia artificial ya hay modelos de neuronas artificiales cuánticas. Su capacidad será de gran utilidad en el desarrollo de modelos y simulaciones en disciplinas como la astronomía, la física y la química. También tendrán aplicaciones en la industria del entretenimiento, por ejemplo en la realización de efectos especiales en el cine.

¿CÓMO FUNCIONA EL ORDENADOR CUÁNTICO?

Un ordenador cuántico es una máquina que, a diferencia de uno convencional, basa su funcionamiento en fenómenos cuánticos. Se trata de fenómenos naturales que no pueden ser explicados por la física convencional; su explicación requiere de una teoría alternativa, la mecánica cuántica, capaz de explicar satisfactoriamente lo que ocurre en la estructura básica de la materia, los átomos. Pese a lo que pudiera parecer, estos fenómenos se manifiestan en nuestra vida diaria. Gracias a ellos podemos explicar, por ejemplo, por qué un objeto es sólido, las propiedades físicas de los materiales o los colores.

Mientras que un ordenador representa los datos como secuencias de unos y ceros, es decir bits, los ordenadores cuánticos, como ya adelantamos anteriormente, lo hacen con qbits. La posibilidad de construir un ordenador cuántico se remonta a 1982, a partir de las investigaciones del célebre físico Richard Feynman, el primer científico en concebir esta clase de ordenadores. En la actualidad su diseño está todavía en sus primeros pasos. Hasta la fecha se han realizado algunos experimentos con unos pocos qbits. También se han diseñado simuladores que emulan esta clase de ordenadores en otros convencionales, pero para que uno convencional pueda ejecutar un algoritmo cuántico, necesita una gran memoria y una gran capacidad de cálculo, además de otras prestaciones de *hardware*. Sin embargo, los experimentos que se pueden realizar son más bien sencillos, lo suficiente para familiarizarse con esta tecnología. Estos simuladores se tienen que limitar a unos pocos qbits, ya que resulta imposible con la tecnología actual almacenar, por ejemplo, 500 qbits.

Pero, ¿cómo funciona un ordenador cuántico? En primer lugar, como ya sabemos, la información se almacena como una secuencia de qbits. A diferencia de un bit, cuyo valor es 0 o 1 —«apagado» o «encendido»—, un qbit puede tener un valor igual a 0, 1 o cualquier otro estado superpuesto, es decir, puede estar simultáneamente apagado y encendido, entre 0 y 1. Un qbit se representa utilizando una notación especial llamada de Dirac, en la que los estados cero y uno se representan como $|0\rangle$ y $|1\rangle$, respec-

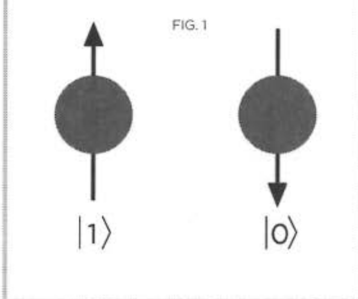


FIG. 1

tivamente. Aunque en la práctica hay varios procedimientos para construir físicamente qbits, simplificaremos este hecho suponiendo que un qbit es una partícula, es decir, algún componente elemental de la materia, por ejemplo, un electrón, que está en estado uno si está orientado hacia arriba, y cero, hacia abajo (figura 1).

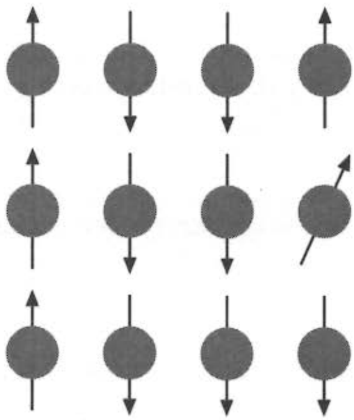
Conviene aclarar también que el sistema de numeración binario (base dos) tiene dos posibles dígitos, representados como 0 o 1, mientras que el sistema decimal (base diez) tiene diez posibles dígitos (0, 1, 2,..., 9). En cada sistema de numeración, cualquier número es una combinación de tales dígitos. Puesto que el sistema binario es el lenguaje interno de los ordenadores electrónicos, la conversión de números de un sistema de numeración a otro es una de las tareas habituales de los programadores. Un método de conversión de un número binario a decimal consiste en asignar una potencia de 2 según su posición, y de derecha a izquierda, a cada dígito binario. Así, por ejemplo, si el número binario es 1011 entonces, de derecha a izquierda, procederemos como sigue: al dígito 1 de la derecha, 2^0 (cuyo valor es la unidad), al siguiente 1, 2^1, al dígito 0, 2^2 y al 1 de la izquierda, 2^3. A continuación, calcularemos la suma de los productos de cada dígito binario por su correspondiente potencia de 2, esto es $1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$, siendo el resultado de la suma el número decimal equivalente, 11 en nuestro caso. En la práctica cuando los números binarios se componen a partir de bloques de cuatro dígitos, el método descrito puede resumirse en la tabla siguiente:

Binario	0000	0001	0010	0011	0100	0101	0110	0111
Decimal	0	1	2	3	4	5	6	7

Binario	1000	1001	1010	1011	1100	1101	1110	1111
Decimal	8	9	10	11	12	13	14	15

A partir de estas especificaciones, ¿cómo representaremos un número con qbits?

FIG. 2



Supóngase, por ejemplo, que deseamos representar el número 9 (figura 2). En sistema binario sería equivalente a 1001, ya que según la expresión $1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$ (recuérdese que 2^0 es 1), el número binario 1001 es igual a 9 en sistema decimal. Por tanto, tendremos que $|9\rangle$ es $|1001\rangle$. ¿Y el número 8? En este caso $|8\rangle$ es $|1000\rangle$. Esto significa que un ordenador cuántico representaría los números 8 y 9 de forma similar a como lo haría un ordenador convencional. Sin embargo, entre otras novedades, el ordenador cuántico puede representar y operar con estados superpuestos, por ejemplo con $|8\rangle + |9\rangle$.

Ahora bien, cuando intentamos averiguar por métodos experimentales «en qué estado superpuesto» está el qbit de todos los estados posibles entre 0 y 1, entonces se manifiesta el principio de interferencia, que consiste en que el qbit, como dicen los físicos cuánticos, se «colapsa». Es decir, el qbit se convierte en un bit clásico, pierde su estado superpuesto y toma un valor igual a 0 o 1. Esto significa que un ordenador cuántico puede realizar sus operaciones según las reglas de la mecánica cuántica, de ahí su potencial, y mostrar al final el resultado al usuario como si de un ordenador convencional se tratase.

Otro de los fenómenos que se dan en los ordenadores cuánticos es el llamado *entrelazamiento cuántico*, una extraña propiedad presente en las partículas de luz, los fotones, entre otras. Según esta propiedad, dos fotones entrelazados se comportarán de tal forma que lo que ocurra en uno de ellos influirá en el otro. Una de las aplicaciones más importantes de este fenómeno en computación cuántica es en el campo de la criptografía, disciplina

en la que Alan Turing hizo grandes contribuciones durante su estancia en Bletchley Park.

Pero pongamos un ejemplo. Sean dos qbits, a los que llamaremos A y B, con estados cero y uno, y que representaremos según la notación como $|0\rangle_A$ y $|1\rangle_B$, respectivamente. Si ambos están entrelazados, entonces utilizaremos el símbolo ⊗, que en matemáticas se usa para designar la operación «producto tensorial», tal y como se muestra a continuación:

$$\frac{1}{\sqrt{2}}\left(|0\rangle_A \otimes |1\rangle_B\right) + \frac{1}{\sqrt{2}}\left(|1\rangle_A \otimes |0\rangle_B\right).$$

En la expresión anterior,

$$\frac{1}{\sqrt{2}}$$

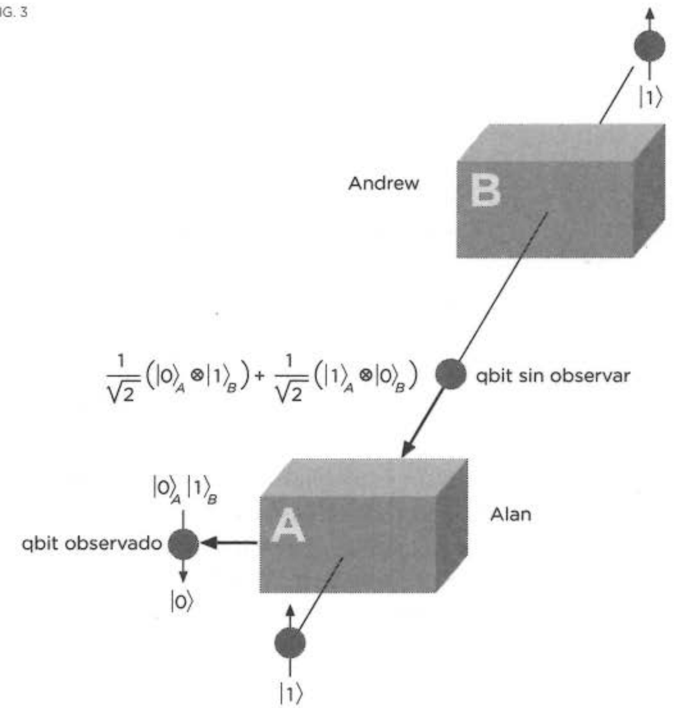
es un valor que procede de aplicar el producto tensorial a un sistema formado por dos qbits. Sin entrar en detalles, puesto que se supone que los qbits se hallan en lo que se conoce como *espacio de Hilbert* —una generalización del espacio euclídeo—, si se eleva al cuadrado este valor, esto es

$$\left(\frac{1}{\sqrt{2}}\right)^2,$$

entonces obtendremos 1/2, que es la probabilidad de medir en un experimento cuántico los estados o de obtener los resultados $|01\rangle$ o $|10\rangle$.

Supóngase ahora que Alan Turing es amigo de Andrew Hodges, su mejor biógrafo, y que el primero puede observar o medir en qué estado está el qbit A, mientras que el segundo puede observar o medir el estado del qbit B. Para hacer el experimento más espectacular, supondremos que Alan y Andrew están en dos habitaciones separadas y ambos disponen de un aparato de laboratorio con el que medir el estado de su qbit. Lo curioso del experimento es que si, por ejemplo, Alan es el primero en observar o medir el estado del qbit que le ha sido asignado (A), podrá obtener como

FIG. 3



resultado, de forma similar al lanzamiento de una moneda (50% de probabilidad para cada evento), que su qbit se encuentre en estado $|0\rangle_A$ o $|1\rangle_A$. Más aún, y este es el aspecto fantástico de la computación cuántica, la observación o medida efectuada por Alan producirá un fenómeno de colapso una vez que haya averiguado el estado de su qbit. El resultado es que para Andrew, que se encuentra en la otra habitación, y puesto que los dos qbits estaban entrelazados, se pierde el carácter aleatorio del experimento. Esto significa que si a continuación Andrew observa o mide el estado de su qbit (*B*), el resultado de su observación estará determinado. Es decir, para Andrew el experimento ya no será equivalente al lanzamiento de una moneda, pues obtendrá en el 100% de las observaciones el resultado inverso al de Alan (figura 3). Por ejemplo,

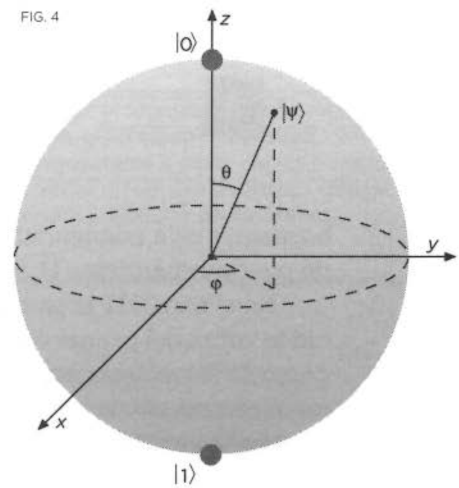
si en el estado entrelazado Alan observó que el qbit A estaba en estado $|0\rangle_A$, entonces la pareja de qbits colapsaría a $|0\rangle_A|1\rangle_B$, mientras que si Alan observó lo contrario, esto es que A estaba en estado $|1\rangle_A$, entonces los qbits se colapsarían a los estados $|1\rangle_A|0\rangle_B$. Es decir, la medida realizada por Alan «alteró» los qbits de tal forma que si por ejemplo obtuvo $|0\rangle_A$ como resultado de su observación, que obtendrá en el 50% de las observaciones realizadas, entonces Andrew obtendrá siempre $|1\rangle_B$ en el 100% de las observaciones.

La utilidad del entrelazamiento cuántico en sistemas de cifrado con fines militares o comerciales es evidente, ya que si dos personas comparten dos objetos entrelazados, la incursión de una tercera persona no autorizada en el sistema alterará cualquiera de los dos objetos, revelando por tanto su presencia. En la actualidad se investiga en sistemas de esta clase basados en el uso de luz polarizada, es decir, en sistemas en los que la luz oscila en un solo plano, de manera que se considera, por ejemplo, que cuando la luz oscila horizontalmente está en estado 0, y en estado 1 si lo hace verticalmente. Por consiguiente, en un ordenador cuántico un qbit podrá estar en los estados $|0\rangle$, $|1\rangle$, en un estado superpuesto entre $|0\rangle$ y $|1\rangle$ o entrelazado con otro qbit, superando así las limitaciones de la máquinas de Turing universales, o si se prefiere, de los ordenadores convencionales.

En segundo lugar, mientras que el *hardware* de un ordenador convencional utiliza puertas AND, OR, etc., uno cuántico se basa en el uso de puertas cuánticas, que operan con qbits y, además, y esto es lo novedoso, sus operaciones son reversibles. Por ejemplo, en una puerta OR de un ordenador convencional, si la salida es 1, la operación realizada es irreversible, lo que significa que es imposible averiguar si las entradas fueron 0 o 1, 1 o 0, 1 o 1. Más aún, la clase de operaciones que un ordenador cuántico puede realizar con qbits es superior a las que pueden realizarse con bits, ya que los estados en que puede encontrarse un qbit pueden representarse como un vector dentro de una esfera, denominada *esfera de Bloch* (figura 4). El programa *blochsphere* simula un qbit en el ordenador, así como las operaciones que pueden ser realizadas.

Además de las operaciones lógicas del álgebra de Boole (AND, OR, etc.), hay otras operaciones con qbits que producen rotaciones del vector sobre los ejes X, Y, Z de la esfera de Bloch. Estas operaciones con qbits son el resultado de aplicar las llamadas puertas cuánticas. Una puerta cuántica es un circuito cuántico que realiza alguna operación sobre uno o más qbits. Por ejemplo, las puertas de Pauli o la de Hadamard permiten hacer rotaciones. Es necesario recordar que al representar un qbit como un vector en la esfera de Bloch, las puertas cuánticas son en realidad matrices que al multiplicarlas por el vector que representa al qbit dan un nuevo vector que será el qbit transformado. Un sencillo ejemplo es la puerta de Pauli de clase x, cuya matriz es:

FIG. 4



Esfera de Bloch. Un qbit está representado por el vector $|\psi\rangle$. Los estados $|0\rangle$ y $|1\rangle$ están en el norte y en el sur de la esfera, y en el resto de la esfera, los estados superpuestos.

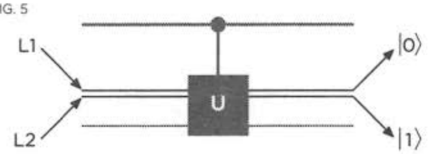
$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

que al ser aplicada a un qbit, su resultado es una rotación en el eje X de la esfera de Bloch, transformando $|0\rangle$ en $|1\rangle$ y $|1\rangle$ en $|0\rangle$. Por ello, en un ordenador cuántico equivale al operador NOT de un ordenador digital. En el caso particular de otra puerta cuántica, la puerta de Hadamard, la rotación del vector que representa al qbit es realizada simultáneamente en los ejes X y Z:

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Otras puertas, como pueden ser por ejemplo CNOT, swap, Toffoli, permiten hacer, entre otras cosas, operaciones controladas con dos o tres qbits, etcétera.

FIG. 5



En tercer lugar, otra de las peculiaridades de un ordenador cuántico es que la transmisión y operaciones con qbits se realizan en paralelo, esto es, simultáneamente por diferentes líneas, por ejemplo por las líneas L1 y L2, y su *hardware* está configurado a partir de la conexión, una tras otra, de puertas cuánticas (U; figura 5).

En el año 2011 la empresa canadiense D-Wave Systems anunció la venta del primer ordenador cuántico comercial, bautizado como *D-Wave One*. Según la empresa, su ordenador disponía de un microprocesador de 128 qbits. Ese mismo año un equipo de investigadores de Estados Unidos, China y Japón anunció que esta clase de ordenadores pueden construirse según el modelo clásico de arquitectura de Von Neumann. En 2012 la empresa IBM anunció que también había realizado avances significativos hacia la construcción de una máquina de estas características. Más de medio siglo después, se repite aparentemente el mismo escenario que tuviera lugar tiempo atrás con ENIAC, Colossus y los otros ordenadores. Sin embargo, esto no es del todo así, ya que la construcción de un ordenador cuántico es un proyecto con tantas dificultades que en esta ocasión investigadores de distintos países han aunado esfuerzos, formando equipos multinacionales y dejando así atrás la competencia entre países. Entre sus aplicaciones, además de la criptografía, se espera que puedan realizarse experimentos de simulación con gran realismo, por ejemplo, las interacciones de los medicamentos en el cuerpo humano, la realización de cálculos en áreas como la física, la química o la astronomía, o su aplicación a problemas matemáticos de cierta envergadura, como es la factorización de grandes números.

Aunque por ahora sea una mera curiosidad, también hay varias versiones cuánticas del Juego de la Vida de Conway. Más aún, en la actualidad han sido propuestos varios modelos de redes neuronales artificiales, cuyas neuronas están simuladas con puertas cuánticas, lo que abre la puerta a futuras investigaciones de lo que podríamos denominar como *inteligencia artificial cuántica*. Otra

> **EMULAR UN ORDENADOR CUÁNTICO**
>
> En la actualidad es posible emular un «pequeño» ordenador cuántico mediante un ordenador convencional. Un ejemplo es el emulador jQuantum, con el que podremos diseñar algunos circuitos elementales a partir de las puertas cuánticas estándar. El programa permite diseñar el registro de datos —puede almacenar hasta 15 qbits—, así como el circuito y ejecutar un algoritmo.

de las aplicaciones es la obtención de números aleatorios que sean «verdaderamente aleatorios», como si tales números hubieran sido obtenidos con un bombo de lotería. De hecho, ya es posible obtener números aleatorios a partir de fenómenos cuánticos a través de Internet (véase www.randomnumbers.info).

EL SUEÑO DE TURING: MAQUINARIA INTELIGENTE AL SERVICIO DE LA VIDA DIARIA

La repentina desaparición de Alan Turing en 1954 no le permitió concluir sus investigaciones en la Universidad de Manchester. Durante su estancia en dicho centro abordó el diseño de modelos de circuitos neuronales con los que estudiar la que él definió como «maquinaria inteligente» en referencia al cerebro humano. En el mismo año de su muerte dos investigadores del Instituto Tecnológico de Massachusetts, Belmont Farley (1920-2008) y Wesley Clark (n. 1927), fueron capaces de lograr con éxito la simulación en ordenador de redes de 128 neuronas capaces de reconocer patrones sencillos tras una fase de entrenamiento. Además observaron que si se eliminaba un 10% de las neuronas, la red no perdía su capacidad de reconocimiento de patrones. El modelo, ciertamente muy elemental, consistía en neuronas conectadas unas con otras al azar, asociando a cada conexión un valor de peso, y el circuito neuronal se comportaba de manera similar a una red de McCul-

loch-Pitts. El entrenamiento de la red neuronal se conseguía de manera parecida a lo que se conoce como regla de Hebb, de tal forma que cuando una neurona estimulaba de forma persistente a otra, aumentaba la eficacia sináptica entre ambas, con lo que crecía el peso en la conexión entre ambas neuronas. En 1956, dos años después de la muerte de Alan Turing, John McCarthy acuñó el término *inteligencia artificial* durante una conferencia acerca de la simulación del comportamiento humano con ordenador, impartida en el Darmouth College, en Estados Unidos. Un año más tarde, en 1957, el psicólogo Frank Rosenblatt (1928-1971) desarrolló el perceptrón, la primera red neuronal artificial con utilidad práctica.

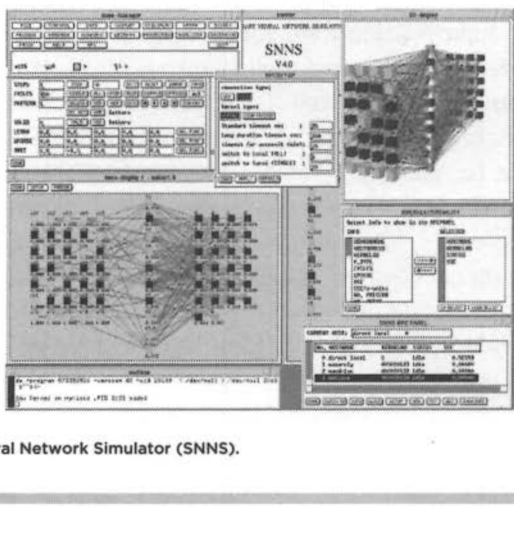
A partir de estas simulaciones surgieron otros modelos de redes neuronales artificiales, por ejemplo, las redes con retropropagación, con las que es posible reconocer letras, números, fotografías, etc., de una manera más eficaz. En la actualidad, tanto las redes sencillas como aquellas con retropropagación son ampliamente utilizadas en la vida diaria, por ejemplo, en la clasificación del correo electrónico para evitar correos no deseados —el *spam*—, en el reconocimiento del habla e imágenes, en el reconocimiento del electroencefalograma (EEG) humano, en el reconocimiento del latido cardiaco del feto para distinguirlo del de la madre, y un largo etcétera. Desde hace años las redes neuronales artificiales han sido «construidas» en circuitos integrados, los llamados *neurochips*, formando parte de tarjetas que pueden ser incorporadas a un ordenador u otra máquina con el fin de desarrollar aplicaciones o sistemas inteligentes en problemas tan variados como los citados anteriormente o, por ejemplo, en problemas de índole financiera. Ha hecho falta que transcurriera más de medio siglo para que las ideas de Turing acerca de la maquinaria inteligente formen parte de nuestra vida cotidiana.

EL ADN Y LA VIDA EN EL ORDENADOR

Hacia el final de su vida, Alan Turing también hizo experimentos pioneros en la simulación de la morfogénesis, esto es, los procesos biológicos que conducen a que un organismo desarrolle su

EMULAR REDES NEURONALES ARTIFICIALES

En la actualidad, los modelos de redes neuronales artificiales tienen numerosas aplicaciones. En general, las redes neuronales utilizan un modelo de organización similar, esto es, las neuronas se organizan por capas (entrada, salida e intermedias u ocultas, si las hubiere) y se conectan entre sí según un criterio inspirado en la biología, por el que las neuronas de una capa se conectan con las de otra. El usuario define en la red cuáles serán los umbrales de activación, la función de activación o transferencia, y otros parámetros del modelo. Sin embargo, pese a que la forma en que se organizan los elementos son muy similares en todas las redes neuronales artificiales, hay un elemento que las distingue unas de otras: el algoritmo o regla de aprendizaje. En inteligencia artificial, el aprendizaje es el proceso por el que una red neuronal cambia su respuesta, o salida, ante una cierta entrada. Ese cambio es el resultado de un ajuste en uno o más de los pesos asociados a las conexiones. Existen multitud de métodos para el ajuste de los pesos de las conexiones en la red con los que esta es «entrenada» en el reconocimiento de patrones (letras, números, fotografías, etc.). En otros casos, la red memoriza directamente el patrón sin necesidad de dicho entrenamiento, es decir, sin que se requiera el ajuste de los pesos de las conexiones. Ni el modelo de McCulloch-Pitts ni el modelo de neurona artificial de Turing eran capaces de exhibir aprendizaje, pues carecían de regla de aprendizaje. Se trataba de modelos con los que se podía emular las puertas AND, OR, etc., es decir, estaban más próximos a una máquina de Turing que a una red de neuronas biológicas. Uno de los mejores programas con los que aprender a través de la experimentación sobre redes neuronales artificiales es el Stuttgart Neural Network Simulator (SNNS).



El Stuttgart Neural Network Simulator (SNNS).

forma, utilizando para tal fin los ordenadores de la Universidad de Manchester. Turing postuló que ciertas sustancias químicas, los morfógenos, así como ciertos procesos físico-químicos, por ejemplo, la difusión, es decir el movimiento de moléculas como el morfógeno, u otros fenómenos como la activación, o promoción, y la inhibición, o represión, eran responsables de los procesos de diferenciación celular, que consiste en las etapas por las que pasa una célula desde el embrión hasta convertirse en el individuo adulto en una célula especializada —muscular, neuronal, etc.—. Por tanto, la idea central de Turing era que en un embrión, las posiciones que ocupan las células aún sin diferenciar, es decir, sin especializar, contienen información «grabada» en los morfógenos con la que se controlará el desarrollo del embrión, el proceso que conducirá a la especialización de sus células hasta llegar a convertirse en un individuo adulto. La genialidad de Alan Turing se manifestó una vez más en esta investigación, ya que predijo la existencia de los morfógenos, que no fueron descubiertos hasta muchos años después.

En los años sesenta el biólogo Lewis Wolpert (n. 1929) redefinió el concepto de morfógeno introducido por Turing, tras descubrir la primera sustancia de estas características, una proteína, en la mosca del vinagre *Drosophila melanogaster*. Los morfógenos, que pueden ser sustancias químicas muy variadas, desde proteínas hasta vitaminas, funcionan controlando los genes, las unidades de la herencia. Sin embargo, puesto que un gen es un fragmento de ADN, su modo de acción no fue entendido hasta el descubrimiento de la estructura del ADN en 1953 por James D. Watson (n. 1928) y Francis Crick (1916-2004), un año antes de la muerte de Alan Turing. En la actualidad, el modelo de Turing de morfogénesis, mediante el que explicó la formación de bandas en la piel de las cebras, ha sido aplicado a otros animales y demostrado experimentalmente. Su modelo ha recibido el apoyo de muchos investigadores en problemas teóricos de la biología, como por ejemplo, Lewis Wolpert (n. 1929) o Hans Meinhardt (n. 1938). Sin embargo, hay investigadores que sostienen que la morfogénesis ocurre de otra forma a la postulada por el científico inglés, que las células siguen un «plan maestro» por el que las células del





ALAN TURINGYEAR



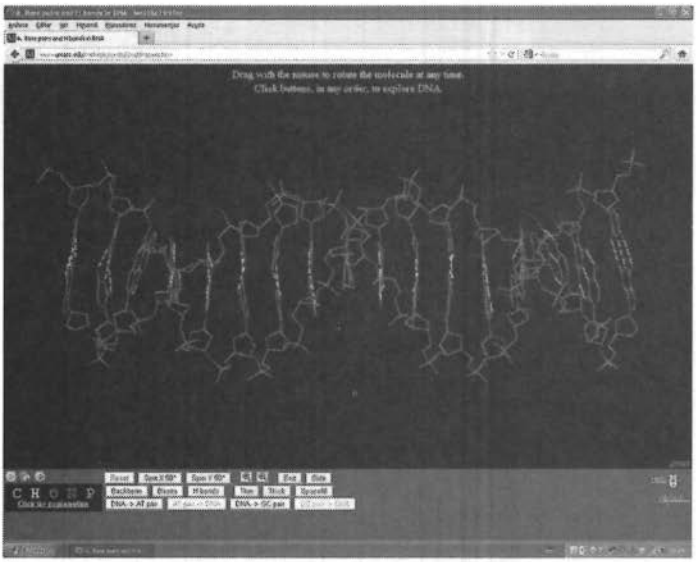
FOTO SUPERIOR IZQUIERDA:
Estatua dedicada a Alan Turing en Sackville Gardens, Manchester, con una manzana en la mano, en referencia al medio que utilizó para suicidarse.

FOTO SUPERIOR DERECHA:
Sello conmemorativo de Alan Turing, puesto en circulación en 2012.

FOTO INFERIOR:
Imagen conmemorativa del centenario del nacimiento de Alan Turing, celebrado el año 2012.

VISUALIZAR EL ADN CON JMOL

Jmol es un visor Java de código abierto con el que es posible visualizar estructuras químicas en tres dimensiones, por ejemplo, compuestos químicos, cristales, materiales y biomoléculas. Uno de los ejemplos más interesantes es la molécula de ADN: se puede rotar, ampliar o reducir, cambiar la clase de representación, etc. El ADN es un polímero con estructura de doble hélice formado por unidades repetitivas, los nucleótidos: son la adenina (A), la citosina (C), la guanina (G) y la timina (T). Los nucleótidos de una hélice se aparean con los de la hélice de enfrente, A con T y G con C, definiendo en cada hélice secuencias, los genes, en las que se almacena información biológica que será transmitida de los individuos de una generación a la siguiente.



El visualizador de Java Jmol.

embrión se irían especializando como consecuencia de una serie de transformaciones explicables a partir de sus propiedades mecánicas. Pueden deformarse, estirarse, etc., hasta especializarse, por ejemplo, como células neuronales, musculares u óseas. Este

conjunto de transformaciones se explican a partir de modelos matemáticos de los fenómenos mecánicos observados en las células. Esta idea, que también utiliza ecuaciones diferenciales, como el modelo de Turing, contó desde hace años con el apoyo de investigadores de gran renombre, como Conrad Waddington (1905-1975), Murray Gell-Mann (n. 1929) o Brian Goodwin (1931-2009).

A partir del descubrimiento del ADN y del diseño de algoritmos para el estudio de la información genética mediante ordenador, nació una nueva disciplina, la *bioinformática*. Pero si el ordenador ha sido y es de gran utilidad en el estudio del ADN, este también ha sido utilizado para el diseño de una nueva clase de ordenadores, cuyo estudio ha dado lugar a la denominada *computación con ADN*. En 1994 Leonard Adleman (n. 1945) realizó experimentos con ADN resolviendo el problema del «camino hamiltoniano», que consiste en encontrar la ruta más corta que pase una única vez por cada lugar, dado un cierto número de lugares, por ejemplo, en el caso de los experimentos de Adleman, siete ciudades. Con estos experimentos se abrió la puerta a que otros investigadores, como Ehud Saphiro (n. 1955), construyeran máquinas de Turing con la molécula del ADN.

EL RECONOCIMIENTO A UN LEGADO

En 1999, la revista *Time* seleccionó a Alan Turing como una de las veinte personas más influyentes del siglo XX. De hecho, desde 1966 la Asociación de Máquinas para la Computación, más conocida por su acrónimo ACM, convoca anualmente el premio Turing, un galardón equivalente al premio Nobel de la Informática. En 2009, Gordon Brown, primer ministro británico en esa época, pidió perdón oficialmente al considerar que el caso de Alan Turing había sido tratado de manera injusta. Sin embargo, en febrero de 2012, una petición de perdón póstuma presentada ante la Cámara de los Lores gracias a una iniciativa que contaba con 23000 firmas fue rechazada.

Para celebrar el 100º aniversario de su nacimiento, 2012 fue elegido como «Año Conmemorativo de Alan Turing», con home-

najes, congresos y reuniones en todo el mundo. El Reino Unido fue el país donde se concentraron el mayor número de acontecimientos, e incluso se emitió un sello conmemorativo con la imagen de Bombe, la máquina con la que Turing y sus colegas descifraron los códigos Enigma, ayudando a su país y los Aliados a ganar la Segunda Guerra Mundial.

Con motivo del centenario, *Scientific American*, una revista de divulgación científica, le dedicó un número especial, titulado *La ciencia después de Alan Turing*. En la actualidad, Alan Turing cuenta con cinco «placas azules», usadas por los ingleses para indicar aquellos edificios en los que nació, vivió o murió un personaje ilustre.

Lecturas recomendadas

ARBIB, M.A., *Cerebros, máquinas y matemáticas*, Madrid, Alianza Universidad, 1987.

BELL, E.T., *Los grandes matemáticos*, Buenos Aires, Losada, 2010.

BOYER, C., *Historia de la matemática*, Madrid, Alianza Editorial, 2007.

COELLO, C.A., *Breve historia de la computación y sus pioneros*, México D.F., FCE, 2003.

CRANE, T., *La mente mecánica. Introducción filosófica a mentes, máquinas y representación mental*, México D.F., FCE, 2008.

ISASI, P., MARTINEZ, P., BORRAJO, D., *Lenguajes, gramáticas y autómatas. Un enfoque práctico*, Madrid, Pearson Educación, 1997.

LAHOZ-BELTRA, R., *Bioinformática. Simulación, vida artificial e inteligencia artificial*, Madrid, Díaz de Santos, 2004.

—: *Turing. Del primer ordenador a la inteligencia artificial*, Madrid, Nivola, 2009.

LEAVITT, D., *El hombre que sabía demasiado*, Barcelona, Editorial Antoni Bosch, 2007.

ODIFREDDI, P., *La matemática del siglo XX: de los conjuntos a la complejidad*, Buenos Aires, Katz Editores, 2006.

PEÑA, R., *De Euclides a Java: Historia de los algoritmos y de los lenguajes de programación*, Madrid, Nivola, 2006.

STEWART, I., *Historia de las matemáticas*, Madrid, Crítica, 2008.

STRATHERN, P., *Turing y el ordenador*, Madrid, Siglo XXI, 1999.

Índice